

Adaptive Hybrid Simulations for Multiscale Stochastic Reaction Networks

Benjamin Hepp,¹ Ankit Gupta,¹ and Mustafa Khammash¹

*Department of Biosystems Science and Engineering (D-BSSE),
ETH Zurich Switzerland*

(Dated: 6 January 2015)

The probability distribution describing the state of a Stochastic Reaction Network evolves according to the Chemical Master Equation (CME). It is common to estimate its solution using Monte Carlo methods such as the Stochastic Simulation Algorithm (SSA). In many cases these simulations can take an impractical amount of computational time. Therefore many methods have been developed that approximate the Stochastic Process underlying the Chemical Master Equation. Prominent strategies are Hybrid Models that regard the firing of some reaction channels as being continuous and applying the quasi-stationary assumption to approximate the dynamics of fast subnetworks. However as the dynamics of a Stochastic Reaction Network changes with time these approximations might have to be adapted during the simulation. We develop a method that approximates the solution of a CME by automatically partitioning the reaction dynamics into discrete/continuous components and applying the quasi-stationary assumption on identifiable fast subnetworks. Our method does not require user intervention and it adapts to exploit the changing timescale separation between reactions and/or changing magnitudes of copy numbers of constituent species. We demonstrate the efficiency of the proposed method by considering examples from Systems Biology and showing that very good approximations to the exact probability distributions can be achieved in significantly less computational time.

Keywords: adaptive hybrid models, stochastic reaction networks, quasi-stationary, adaptive piecewise deterministic markov processes

I. INTRODUCTION

Chemical reaction networks, where a finite number of molecular species interact with each other through a fixed number of reaction channels, are an important tool for modeling many biochemical systems. The reaction dynamics is invariably *noisy* due to the discrete nature of molecular interactions which causes the timing of reactions to be random. It is known that this noise can be neglected for systems with high copy-numbers of all the species, and the dynamics can be modeled deterministically through a set of ordinary differential equations¹. However many biological systems involve molecular species with low copy-numbers and hence the randomness in the dynamics can have a significant impact on the properties of the system. This has been demonstrated in a number of biological systems, such as gene expression², piliation of bacteria³ and polarization of cells⁴, as well as in synthetic biological circuits such as the Toggle Switch⁵ and the Repressilator⁶.

To understand the role of noise and its effects, reaction networks are commonly modeled as stochastic processes with Markovian dynamics where the state represents the copy-numbers of the molecular species⁷. The dynamics of the distribution of a Markov process representing the reaction network evolves according to the Chemical Master Equation (CME) which is a set of ordinary differential equations (1). The size of this system is equal to the number of elements in the state space, which is typically infinite, making the task of solving the CME practically impossible for most interesting systems. One can solve a projection of the CME onto a finite subspace, but this only works for small systems⁸. Other methods are developed that allow the direct solution of the CME for specific classes of networks⁹.

If the CME cannot be solved directly, one usually resorts to Monte Carlo methods to generate trajectories of the underlying stochastic process and approximate the probability distribution through a large number of simulations. These simulations can be performed using Gillespie’s Stochastic Simulation Algorithm (SSA) or its variants¹⁰, that generate exact sample paths by taking into account the firing of each reaction within the simulation time-period. Several biological systems have reactions spanning a wide range of timescales, either due to variation in the magnitude of the rate constants or variations in the copy-numbers of the constituent species. If the system has reactions with fast timescales, then simulation schemes like SSA can take an impractical amount of time^{11,12}.

To handle this problem, approximate schemes such as τ -leaping methods have been de-

veloped, that perform multiple reactions at each step¹³. Such methods are very efficient in simulating many systems with high reaction rates and many enhancements have been proposed to automatically select a proper leap-size^{14,15} or deal with networks where the dynamics is “stiff” and some reactions occur on a very fast timescale¹⁶.

In most biological applications, one is interested in computing the probability distributions described by a CME. If we can capture the necessary stochasticity using approximate sample paths that are easier to simulate, then we can efficiently obtain a close approximation to the solution of a CME. For this to work we need to identify and conserve the important sources of stochasticity while discarding the insignificant stochastic effects. The dynamical law of large numbers suggests that stochastic effects are less important for a species if the copy-numbers are large¹⁷. For example, consider a simple gene-expression network, where DNA is transcribed into mRNA and mRNA is translated into proteins. One might expect that the stochasticity of transcription and translation of low copy-number DNA and mRNA can be important but the stochasticity of the high-copy number proteins for downstream processes to be unimportant. If the copy-numbers of *all* the species are large, then under a suitable scaling of rate constants, the process describing the species concentrations¹⁸, converges in the limit of infinite copy-numbers to the reaction rate equations that correspond to the deterministic model of the reaction network¹⁷. Instead of scaling the copy-numbers of all the species uniformly and taking the limit, one can construct processes where the species are partitioned into a set described by concentrations and a set described by copy-numbers. Such a process has been shown to converge in the infinite copy-number limit for a number of examples^{19,20}. The limiting processes in these cases are hybrid processes, that combine both deterministic and Markovian dynamics and, can be called Piecewise Deterministic Markov processes (PDMP)²¹.

This idea has been exploited in various hybrid schemes^{22–33}, which mostly differ in the way that the partitioning is performed and the way in which the hybrid system is simulated. The partitioning of the species and/or reactions is either done manually or certain threshold-values are chosen for different properties of the system, e.g. the copy-numbers of species^{23,24,29}, the propensities of reactions^{22,33}, the approximated copy-number fluctuations of species²⁵ or a combination of species copy-numbers and reaction propensities^{26,28,30–32}. For the simulation, various combinations of SSA, τ -leaping, stochastic differential equations (SDE) and ordinary differential equations (ODE) are possible. Most methods combine the

SSA approach with a SDE or an ODE approach^{22,23,26,28–30} but there are also methods that introduce a regime for τ -leaping^{31–33}. Another important point for the simulation is the way in which the timings of discrete events is approximated. When integrating the system with τ -leaping, SDE or ODE approach, the propensities of the discrete reactions change with time. Incorporating these time-dependent propensities into the sampling of the discrete event times improves the accuracy of these methods. As soon as the criteria for the partitioning of the system is defined, one can establish an adaptive scheme to repartition the system when necessary. This is important for situations where the orders of magnitudes of the species copy-numbers change significantly over time, a feature found in many important reaction networks from Systems Biology such as transcriptional bursting in gene expression³⁴ or the synthetic Repressilator circuit⁶. How to properly decide when a repartitioning is necessary remains an open issue. We refer the reader to³⁵ for a more comprehensive overview.

The dynamical law of large numbers can be exploited to approximate the dynamics when reaction timescales differ due to variations in the copy-numbers of the constituent species. As mentioned before, timescale separation between reactions can also be caused due to differences in the magnitudes of the rate constants¹². Some hybrid schemes simplify the dynamics in this situation by applying the quasi-stationary assumption for the dynamics of fast subnetworks²⁷. This approach is justified in situations where the exact dynamical details of fast subnetworks are unimportant, but only the distributions of the fast species are required to correctly estimate the propensities of the slow reactions. For example, in many biochemical systems the quasi-stationary assumption can be applied to fast subnetworks consisting of enzyme-substrate interactions and exact simulations of the fast dynamics can be avoided, significantly reducing the computational effort^{12,36}.

In this paper we propose a new hybrid scheme to estimate solutions of the CME corresponding to a stochastic reaction network, which may exhibit multiple reaction timescales due to variation in both the magnitudes of rate constants and copy-numbers of the constituent species. Our method relies on the rigorous mathematical framework recently provided by Kang et al.¹⁹ to simplify the dynamics for such networks. In this framework they introduce scaling parameters for rate constants as well as species copy-numbers and give formal criteria for convergence to a PDMP. Using Linear Programming we select these scaling parameters to fulfil their criteria and then simulate the limiting PDMP. When the magni-

tudes of species copy-numbers change significantly, our method adapts itself by recomputing the appropriate scaling parameters and continues the simulation with the corresponding PDMP. In other words, our method dynamically stitches together several PDMPs to account for the variations in the magnitudes of the species copy-numbers, and the resulting variations in the timescales of various reactions. The method can also automatically identify fast subnetworks and apply the correct quasi-stationary assumption whenever possible.

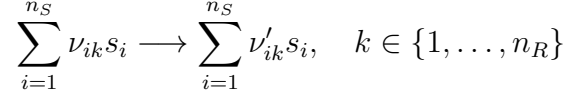
The paper is organized as follows. In Section II we give a mathematical background and introduce the necessary formalism for the rest of the paper. In Section III we elaborate on the algorithm and provide implementation details of our adaptive hybrid scheme. In Section IV we suggest a possible combination of our scheme with τ -leaping schemes. In Section V we compare our adaptive hybrid scheme with SSA and a fixed PDMP scheme (without dynamic repartitioning) for three different examples. We also make a comparison with an existing hybrid scheme (which also dynamically repartitions). Finally, in Section VI we conclude and give an outlook for future work. Additionally, in Appendix A we explain the application of the quasi-stationary assumption and in Appendix B we provide a mathematical justification for the correctness of our adaptive hybrid scheme.

II. MATHEMATICAL PRELIMINARIES

The goal of this section is to present the relevant mathematical background for this paper. We start by describing how the dynamics of a reaction network can be modeled as a continuous time Markov chain. Such models are called *Stochastic Reaction Networks* (SRNs) in this paper. We then motivate and describe the multiscale modeling framework of Kang et al.¹⁹, which explicitly accounts for the variation in reaction timescales by considering both copy-number scales and the differences in magnitudes of rate constants. We also present the limit theorem proved in Kang et al.¹⁹ which shows that under suitable assumptions, the dynamics of the underlying stochastic process is well-approximated by a *Piecewise Deterministic Markov Process* (PDMP) which can be far easier to simulate than the original dynamics. Finally we end this section with a brief discussion on how the simplified dynamics can be obtained even in situations where the required assumptions for the limit theorem fail. In such cases it is often possible to first apply the quasi-stationary assumption and then use the PDMP approximation as before.

Stochastic Reaction Network (SRN)

Consider a well-stirred chemical system with n_S species $\{s_1, \dots, s_{n_S}\}$ which interact according to n_R reaction channels of the form



where ν_{ik} and ν'_{ik} denote the number of molecules of the i -th species that are consumed and produced by the k -th reaction. In the stochastic setting, the reaction dynamics can be represented as a continuous time Markov chain, whose state at any time t is just the vector $x = (x_1, \dots, x_{n_S})$ of copy-numbers of all the species. When the state is x , the k -th reaction occurs after a random time that is exponentially distributed with rate $\lambda'_k(x)$, assuming no other reactions occur first. The function λ'_k is called the *propensity function* for the k -th reaction. We assume that the network only consists of elementary reactions³⁷ and the propensity functions follow mass action kinetics³⁸, as shown in Table I.

Reaction type	Consumption stoichiometry	Propensity function
Constitutive	$\nu_k = 0$	$\lambda'_k(x) = \kappa'_k$
Monomolecular	$\nu_k = e_i$	$\lambda'_k(x) = \kappa'_k x_i$
Bimolecular	$\nu_k = e_i + e_j, i \neq j$	$\lambda'_k(x) = \kappa'_k x_i x_j$
Bimolecular	$\nu_k = 2e_i$	$\lambda'_k(x) = \frac{1}{2} \kappa'_k x_i (x_i - 1)$

TABLE I. Description of mass action kinetics. Here e_i is the i -th standard basis vector in \mathbb{R}^{n_S} .

The CME corresponding to this SRN is given by

$$\frac{dp(x, t)}{dt} = \sum_{k=1}^{n_R} \lambda'_k(x - \xi_k) p(x - \xi_k, t) - \sum_{k=1}^{n_R} \lambda'_k(x) p(x, t) \quad (1)$$

where $\xi_k = \nu'_k - \nu_k$, $\nu_k = (\nu_{1k}, \dots, \nu_{n_S k})$ and $\nu'_k = (\nu'_{1k}, \dots, \nu'_{n_S k})$. In this equation, $p(x, t)$ is the probability $\mathbb{P}(X(t) = x)$, where $\{X(t) : t \geq 0\}$ is the Markov process describing the reaction dynamics. This process satisfies the following random time-change representation³⁹

$$X(t) = x(0) + \sum_{k=1}^{n_R} Y_k \left(\int_0^t \lambda'_k(X(s)) ds \right) \xi_k \quad (2)$$

where $\{Y_k : k = 1, \dots, n_R\}$ is a family of independent unit rate Poisson processes.

In order to understand the behavior of a SRN it is important to compute the distributions that evolve according to the CME. However note that the CME consists of as many equations as the number of elements in the state space of the reaction dynamics, which is typically very large or infinite. This makes CME practically impossible to solve in most cases and therefore its solution is usually estimated using Monte Carlo methods such as Gillespie’s SSA¹⁰ which generates exact paths of the underlying stochastic dynamics. As mentioned before, many biological networks contain reactions firing in several different timescales, and in such situations these exact Monte Carlo methods can be computationally intractable due to the “stiffness” of the system¹⁶. In these cases, it is sometimes possible to approximate the solutions of a CME using a simplified description of the dynamics, which is obtained by preserving the important sources of stochasticity in the dynamics while discarding the rest. One way to obtain such simplified dynamics is to use the multiscale modeling framework of Kang et al.¹⁹ and show that under certain conditions, the original dynamics is well-approximated by a PDMP²¹ which is often much easier to simulate. We now describe this multiscale modeling framework.

Multiscale models of Stochastic Reaction Networks

A multiscale SRN is characterized by reactions firing at many different timescales. This variation in timescales could be both due to variation in copy-number scales as well as variation in the magnitudes of the rate constants. Kang et al.¹⁹ present a framework for separating both these sources of timescale variation, and obtain PDMP approximations of the dynamics under certain conditions. To apply this framework we first select a large positive number N_0 that reflects the typical copy-number of a species which is considered abundant in the network. Note that the PDMP approximation that we later describe would ignore the stochasticity in copy-numbers of species if they are of order N_0^α for some $\alpha > 0$. Hence the accuracy of this approximation and our method which relies on it, crucially depends on the choice of N_0 .

As an example, consider a gene expression network where the number of genes is usually 1 or 2, the number of mRNAs is in the order of tens and the number of proteins is in the order of thousands. To capture the bursting behavior of stochastic gene expression³⁴ we want to keep the discreteness of the genes and the mRNAs. However, the discreteness

of reaction channels driven by proteins can often be neglected, as the fluctuations of the protein numbers are rather small compared to the total protein number. Thus we want to approximate the number of proteins as a continuous quantity and reaction channels driven by proteins as occurring continuously. For this purpose we can choose $N_0 = 1000$.

From now on, we refer to a quantity as order 1 (denoted as $O(1)$) if it remains bounded as N_0 gets larger. Assume that the rate constants κ'_k have different orders of magnitude. For each reaction $k = 1, \dots, n_R$ we pick a $\beta_k \in \mathbb{R}$, such that $\kappa_k = N_0^{-\beta_k} \kappa'_k$ is $O(1)$. We can interpret β_k as the timescale of the k -th reaction, when all the reactants have copy-numbers of order one. To account for the variation in the copy-number scales of species we introduce scaling parameters $\alpha_i \geq 0$ for $i = 1, \dots, n_S$ such that $Z_i^{N_0}(t) = N_0^{-\alpha_i} X_i(t)$ is $O(1)$, at least for t close to 0. The choice of scaling parameters α_i -s and β_k -s is not arbitrary as they must satisfy certain conditions that we later describe. In Section III we will present an automatic scheme for selecting these parameters in a suitable way.

Using these scaling parameters α_i -s and β_k -s, we can derive the scaled process $\{Z^{N_0}(t) = (Z_1^{N_0}(t), \dots, Z_{n_S}^{N_0}(t)) : t \geq 0\}$ from the original process $\{X(t) : t \geq 0\}$. Replacing N_0 by N we obtain a family of processes $\{Z^N(t) : t \geq 0\}$ parametrized by N . If we can show that the process Z^N converges in distribution to another process Z as $N \rightarrow \infty$, then for large values of N_0 , $Z_i^{N_0}(t) \approx Z_i(t)$ and hence $X_i(t) \approx N_0^{\alpha_i} Z_i(t)$ for all i . This gives us a way to approximate our original process X with another process Z which can be much simpler to simulate. In the classical thermodynamic limit¹⁷, N_0 is the volume of the system, $\alpha_i = 1$ for each i and $\beta_k = 1, 0, -1$ depending on whether the k -th reaction is constitutive, monomolecular or bimolecular. In this case the limiting process Z corresponds to the deterministic model of the reaction network and its evolution is given by a systems of ODEs called the reaction rate equations. However for many biochemical networks this classical scaling is not suitable, but a general scaling prescribed by (α_i, β_k) can be used. We now present the main convergence result on which our method is based.

Convergence to a PDMP

Under the scaling prescribed by (α_i, β_k) , the random time-change representation of the process $\{Z^N(t) = (Z_1^N(t), \dots, Z_{n_S}^N(t)) : t \geq 0\}$ is given by

$$Z_i^N(t) = Z_i^N(0) + N^{-\alpha_i} \sum_{k=1}^{n_R} Y_k \left(\int_0^t N^{\beta_k + \alpha \cdot \nu_k} \lambda_k^N(Z^N(s)) ds \right) \xi_{ik} \quad (3)$$

where $Z_i^N(0) = N_0^{-\alpha_i} X_i(0)$ and $\alpha \cdot \nu_k$ denotes the dot product of vectors $\alpha = (\alpha_1, \dots, \alpha_{n_S})$ and ν_k . The λ_k^N -s are defined by

Reaction	$\emptyset \rightarrow \dots$	$S_i \rightarrow \dots$	$S_i + S_j \rightarrow \dots$	$2S_i \rightarrow \dots$
$\lambda_k^N(z)$	κ_k	$\kappa_k z_i$	$\kappa_k z_i z_j$	$\kappa_k z_i (z_i - N^{-\alpha_i})$

with $\kappa_k = N_0^{-\beta_k} \kappa'_k$. In Theorem 4.1. Kang et al.¹⁹ prove that the process Z^N converges to a well-behaved process Z as $N \rightarrow \infty$, if (α_i, β_k) satisfy

$$\alpha_i \geq \beta_k + \alpha \cdot \nu_k \quad \text{for each } i, k \text{ with } \xi_{ik} \neq 0. \quad (4)$$

Moreover the limiting process Z satisfies

$$Z(t) = x(0) + \sum_{k \in R_D} Y_k \left(\int_0^t \lambda_k(Z(s)) ds \right) \xi_k + \sum_{k \in R_C} \left(\int_0^t \lambda_k(Z(s)) ds \right) \xi_k, \quad (5)$$

where $\{Y_k : k \in R_D\}$ is a family of independent unit rate Poisson processes, and reaction subsets $R_C, R_D \subset \{1, \dots, n_R\}$ are defined by

$$\begin{aligned} R_C &= \{k : \alpha_i = \beta_k + \alpha \cdot \nu_k > 0 \text{ for all } i \text{ such that } \xi_{ik} \neq 0\} \text{ and} \\ R_D &= \{k \notin R_C : \alpha_i = \beta_k + \alpha \cdot \nu_k = 0 \text{ for all } i \text{ such that } \xi_{ik} \neq 0\}. \end{aligned} \quad (6)$$

The limiting process Z described by (5) is essentially a *Piecewise Deterministic Markov Process* (PDMP) because a part of the dynamics, given by reactions in R_C evolves continuously (as in an ODE), while another part of the dynamics, given by reactions in R_D evolves discretely like a jump Markov process similar to our original process X . Note that in general $R_C \cup R_D \neq \{1, \dots, n_R\}$ and all reactions not in $R_C \cup R_D$ do not contribute to the dynamics of the limiting process Z .

With the scaling parameters (α_i, β_k) we also define subsets of species by

$$\begin{aligned} S_C &= \{i \in \{1, \dots, n_S\} : \alpha_i > 0\} \\ S_D &= \{1, \dots, n_S\} \setminus S_C. \end{aligned} \quad (7)$$

The species in S_C change due to reactions in R_C and therefore they are measured by “concentrations” and evolve continuously, whereas the species in S_D will have discrete state values given by their copy-number. The dynamics of the limiting process Z can then be written as

$$\begin{aligned} Z_C(t) &= x_C(0) + \sum_{k \in R_C} \left(\int_0^t \lambda_k(Z(s)) ds \right) \xi_k \\ Z_D(t) &= x_D(0) + \sum_{k \in R_D} Y_k \left(\int_0^t \lambda_k(Z(s)) ds \right) \xi_k \end{aligned} \quad (8)$$

where $Z_C(t)$ and $Z_D(t)$ are vectors with entries $Z_i(t)$ for $i \in S_C$ and $i \in S_D$ respectively. The definitions of $x_C(0)$ and $x_D(0)$ are similar.

To simulate such a PDMP, the dynamics of the reactions in R_C are basically an *Ordinary Differential Equation* (ODE) and the reactions in R_D can modify its vector field at random times. The ODE can be solved with established and efficient ODE solvers and for large propensities of the reactions channels in R_C , this can be computationally much cheaper than simulating the discrete individual reactions as Gillespie’s SSA¹⁰. Therefore the speed enhancements that our method obtains (by simulating a PDMP (5)) in comparison to SSA, is directly proportional to the number of reactions treated continuously. More details on simulating a PDMP are provided in section III.

For the above convergence result to hold, we need to find scaling parameters (α_i, β_k) such that (4) is satisfied. Note that such a choice always exists as we can simply set each α_i and β_k to 0 to satisfy these constraints. However with such a trivial choice of scaling parameters, the limiting process Z is same as the original process X and hence we do not obtain any computational advantage in simulating the limiting process. In Section III we describe how these parameters can be automatically chosen using Linear Programming in such a way, that the limiting dynamics is computationally much easier to simulate. The key idea is to treat as many reactions continuously as possible without violating the constraints that imply the PDMP convergence result. We also discuss how they can be properly adapted if the copy-number scales of species vary significantly with time.

Applying the quasi-stationary assumption

Consider the situation where certain species have low copy-numbers but their dynamics is affected by reactions with large rate constants κ'_k . In this situation, an appropriate copy-

number scaling parameter α_i for such species would be close to 0, since copy-numbers are small, while an appropriate rate constant scaling parameter β_k for these reactions would be strictly positive. As a consequence, for these low-copy species with *fast* dynamics (or simply *fast* species), the constraints (4) will fail to hold for scaling parameters (α_i, β_k) that truly represent the dynamics. If we use our automatic parameter selection procedure given in Section III, then all the reactions involved in changing these *fast* species will have their β_k -s set to a negative or a small positive value even though their rate constants κ'_k -s are large. Hence our method will not be able to treat these fast reactions continuously and therefore it will suffer from the same stiffness problems as Gillespie’s SSA¹².

To remedy this problem we use another result in Kang et al.¹⁹, which shows that even with such *fast* species in the network, well-behaved limits can be obtained for certain projections of the process Z^N . Such projections correspond to “reduced” reaction networks which only consist of those species that satisfy (4). These reduced models can also consist of linear combinations of *fast* species that satisfy a constraint analogous to (4). To obtain these reduced models along with their limits, one needs to find subnetworks of fast reaction channels where the quasi-stationary assumption can be applied. Assume that we have such a subnetwork whose internal dynamics is much faster than the surrounding dynamics. If the subnetwork dynamics converges to a stationary distribution then we can use this distribution to estimate the propensities of reactions emanating from this subnetwork^{27,40,41}. In this approach we assume that the species influenced by these fast reaction channels are always at stationarity and hence we do not capture their transient dynamics, thereby saving a lot of computational time^{12,36}. The application of quasi-stationary approximation is also called averaging in the mathematical literature. We discuss this approach in greater detail in Appendix A.

III. IMPLEMENTATION

The aim of this section is to provide full implementation details of our method for approximating the solution of a CME. The presented algorithms compute a single sample path of the dynamics. The solution of a CME at a certain time t can be approximated by generating several sample paths and computing the histogram of the sample-path values at time t . We start by introducing the well-known scheme for simulating a PDMP, where the partitioning

of reactions/species into discrete/continuous subsets is fixed. We then describe our adaptive hybrid scheme, where this partitioning changes with time due to the variation in the orders of magnitude of species copy-numbers. This partitioning is based on the scaling parameters α_i, β_k , mentioned before, and we describe how these parameters can be appropriately chosen using Linear Programming. In both these approaches (Fixed and Adaptive PDMP) computational time can be saved by applying the quasi-stationary assumption on fast subnetworks consisting of discrete species and discrete reactions. We explain this procedure and indicate how it can be integrated into our method.

The inputs that are the same to all the algorithms are given by:

t₀: initial time of simulation

t_f: final time of simulation

z₀: initial state vector of simulation

ν_{ik} **for** $i \in \{1, \dots, n_S\}$ **and** $k \in \{1, \dots, n_R\}$: the consumption stoichiometries

ν'_{ik} **for** $i \in \{1, \dots, n_S\}$ **and** $k \in \{1, \dots, n_R\}$: the production stoichiometries

κ'_k **for** $k \in \{1, \dots, n_R\}$: the unscaled reaction rate constants

$N_0 \in \mathbb{N}$: copy-number scale considered to be large (default value $N_0 = 1000$)

The adaptive algorithm needs two additional inputs:

μ : continuous copy-number scale threshold (default value $\mu = 0.5$)

η : adaptation scale threshold (default value $\eta = 0.9$)

These are described in more detail in the following sections.

We will use the following notation throughout this section:

P: total number of firings considering all discrete reactions occurring from t_0 until t_f

t_r(p): time of the p -th firing considering all discrete reactions for $p = 1, \dots, P$. We define

$$t_r(0) = t_0 \text{ and } t_r(P+1) = \infty.$$

R_D(t): the set of discrete reaction channels at time t

R_C(t): the set of continuous reaction channels at time t

S_D(t): the set of discrete species at time t

$\mathbf{S}_C(\mathbf{t})$: the set of continuous species at time t

$\mathbf{z}(\mathbf{t})$: the state vector at time t

Note that the entries in the state vector $z(t)$ corresponding to $S_D(t)$ are non-negative integers representing the species copy-numbers whereas the entries corresponding to $S_C(t)$ are non-negative real numbers representing the scaled species copy-numbers.

Simulation of Fixed Piecewise Deterministic Markov Processes

Consider a typical PDMP model where the continuous and discrete reaction subsets, given by $R_C(t)$ and $R_D(t)$ respectively, and the induced species partition $S_C(t)$ and $S_D(t)$ do not depend on the time t . The usual algorithm to simulate such models is to evolve the part of the model that is described deterministically until the next discrete reaction occurs. When a discrete reaction occurs, the copy-numbers are updated accordingly. This is repeated until the end-timepoint of the simulation is reached. To evolve the deterministic system we use existing ODE solvers.⁴²

The time increment $\Delta t_p = t_r(p) - t_r(p-1)$ between the $(p-1)$ -st and p -st reaction is defined by the stopping condition

$$\int_{t_r(p-1)}^{t_r(p-1)+\Delta t_p} \sum_{k \in R_D} \lambda_k(z(t)) = u_p \quad (9)$$

where $u_p = -\log(q_p)$, $q_p \sim \mathcal{U}[0, 1]$ and $\mathcal{U}[0, 1]$ is the uniform distribution over $[0, 1]$. At time $t_r(p)$ the k -th reaction in R_D fires with probability

$$\frac{\lambda_k(z(t_r(p)))}{\sum_{m \in R_D} \lambda_m(z(t_r(p)))} \quad (10)$$

For time $t_r(p-1) < t < t_r(p)$, only the states of continuous species will change according to the ordinary differential equation given by

$$\frac{d}{dt} z(t) = \sum_{k \in R_C} \lambda_k(z(t)) \xi_k \quad (11)$$

We start the simulation at time $t = t_0$ and $z(t_0) = z_0$ and generate a random variable u_1 with distribution $\mathcal{U}[0, 1]$. We evolve the state $z(t)$ according to (11) until the first discrete reaction occurs at time $t_r(1) = t_0 + \Delta t_1$, defined by (9). We sample the reaction k from R_D according to (10) and set $z(t_r(1)) = z(t_r(1)-) + \xi_k$. These steps are repeated until we

reach the final time t_f . Below we give an algorithmic description of the procedure we just described.

Algorithm 1: Fixed PDMP

```

1.  $t \leftarrow t_0, z(t_0) \leftarrow z_0, p \leftarrow 1$ 
2. while  $t < t_f$  do
3.   % Set up random variable for the stopping condition (9)
4.    $u_p \leftarrow -\log(q_p)$  where  $q_p \sim \mathcal{U}[0, 1]$ 
5.   evolve
6.    $z$  according to  $\frac{d}{dt}z(t) = \sum_{k \in R_C} \lambda_k(z(t)) \xi_k$ 
7.   until  $t = t_f$  or  $t = t_r(p)$  according to (9)
8.   if  $t = t_r(p)$  then
9.     % Determine which discrete reaction occurs at  $t$  and
       update state vector
10.    Sample  $r$  with  $\Pr(r = k) \sim \lambda_k(z(t)), k \in R_D$ 
11.     $z(t) \leftarrow z(t-) + \xi_r$ 
12.     $p \leftarrow p + 1$ 
13.  end if
14. end while

```

Adaptive Simulation of Piecewise Deterministic Markov Processes

For SRNs that show big variation in copy-numbers over time, it may be impossible to find a fixed partitioning of the reaction and species sets that works well until the final time t_f . This motivated us to extend the fixed PDMP algorithm into an adaptive scheme, where the partitioning can change with time, depending on the copy-number scales of different species. The general idea is to define bounds for the copy-numbers in a suitable manner and upon leaving these bounds the partitioning of the reaction and species sets is updated according to the current copy-number scales.

The copy-number bounds are defined by parameters $\mu \geq 0$ and $\eta \geq 0$. For a discrete species the upper bound is N_0^μ , while for a continuous species the lower and upper bounds are $N_0^{-\eta}$ and N_0^η respectively. Thus the parameter μ is used to decide when a species can be

considered continuous and the parameter η gives the range of re-scaled copy-numbers where no adaptation is performed. We use $\mu = 1$ and $\eta = 0.9$ as default parameter values. Only in the case of too few or too many adaptations do we advise to try different values for η . Note that even though the performance of our algorithm depends on the choice of μ and η , this dependence is relatively small. For each $i \in \{1, \dots, n_S\}$ define a set

$$B_i^{N_0} = \begin{cases} [0, N_0^\mu] & \text{if } i \in S_D \\ [N_0^{-\eta}, N_0^\eta] & \text{if } i \in S_C \end{cases} \quad (12)$$

and update the partitioning of the reactions and species if the value of x_i leaves the set $B_i^{N_0}$.

We implemented our adaptive simulation scheme both with a fixed time step and an adaptive time step. Here we describe the fixed time step algorithm with a time step dt : We define a subset $R_Q \subseteq R_D$ of discrete reactions that influence the continuous dynamics

$$R_Q = \{k \in R_D : \xi_{ik} \neq 0 \text{ and } \nu_{il} \neq 0 \text{ for some } l \in R_C\} \quad .$$

After an integration step of the continuous dynamics has been performed, the next time t_R of a discrete reaction can be computed by using the sum of discrete propensities w computed in the subroutine `integrateStep`. This is repeated until either the next discrete reaction time t_R would exceed the current time of the integration $t_N = t + dt$, or the discrete reaction belongs to R_Q . If the discrete reaction belongs to R_Q then the current time of integration t_N is set to the discrete reaction time t_R .

If the copy-number bounds in relation (12) become invalid, an adaptation procedure is performed where the scaling parameters α_i, β_k are recomputed, the state of the system is scaled accordingly and the averaging of fast subnetworks is performed, if possible. The adaptation and averaging procedures are described in more detail in the following sections.

Algorithm 2: Adaptive PDMP

1. $t \leftarrow t_0, z \leftarrow z_0, w \leftarrow 0$
2. $u \leftarrow -\log(q)$ where $q \sim \mathcal{U}[0, 1]$
3. **while** $t_0 < t_f$ **do**
4. $t_N, [z_N, w_N] \leftarrow \text{integrateStep}(t, [z, w])$
5. $\Delta z = 0$

```

6.  while  $w_N \geq u$  do
7.     $t_R \leftarrow \text{findRoot}(t, w - U, t_N, w_N - U)$ 
8.     $[z_R, w_R] \leftarrow \text{interpolate}(t_R, t, [z, w], t_N, [z_N, w_N])$ 
9.     $r \leftarrow k$  with probability  $p_k \propto \lambda_k(z_R + \Delta z)$ 
10.    $u \leftarrow u - \log(q)$  where  $q \sim \mathcal{U}[0, 1]$ 
11.   if  $r \in R_Q$  then
12.      $z_R \leftarrow z_R + \xi_r$ 
13.      $t_N \leftarrow t_R, z_N \leftarrow z_R$ 
14.     Goto step 21
15.   else
16.      $\Delta z \leftarrow \Delta z + \xi_r$ 
17.      $t \leftarrow t_R, z \leftarrow z_R$ 
18.      $w_N \leftarrow w_R + \sum_{k \in R_D} \lambda_k(z_R + \Delta z) \times (t_N - t_R)$ 
19.   end if
20. end while
21.  $t \leftarrow t_N, z \leftarrow z_N + \Delta z, w \leftarrow w_N$ 
22. for each  $i \in S_D \cup S_C$  do
23.   if  $z_i$  violates (12)
24.     Perform adaptation (Algorithm 3)
25.     Goto step 3
26.   end if
27. end for
28. end while

```

Computation of scaling parameters

Our method relies on computing the PDMP approximation based on the framework by Kang et al.¹⁹ described before. For this we need to pick the scaling parameters α_i -s and β_k -s such that the PDMP approximation is justified. Moreover, to enhance the computational efficiency, we would also like to maximize the number of reactions that can be treated continuously. We achieve these two objectives by formulating the conditions given by (4)

as constraints of a linear program that maximizes a suitably chosen weighted sum of the α_i -s and β_k -s. Once these parameters have been chosen, the result of Kang et al.¹⁹ shows that the PDMP approximation must be partitioned according to (6). However such a strict partitioning scheme is often inappropriate from the standpoint of numerical simulations, because the α_i -s and β_k -s are real numbers. Hence we relax this partitioning criteria by introducing another parameter δ (with $\delta = 1$ as the default value) which decides when a reaction is considered continuous or discrete. Formally, we define

$$R_C(t) = \{k : \alpha_i > \delta \ \forall i \text{ s.t. } \xi_{ik} \neq 0\} \quad \text{and} \quad R_D(t) = \{k : \exists i : \alpha_i \leq \delta, \xi_{ik} \neq 0\}. \quad (13)$$

The partitioning of the species set into discrete and continuous subsets $S_D(t)$ and $S_C(t)$ is defined in Section II.

We now elaborate on the linear program that is solved to find the scaling parameters at time t . For each $i \in \{1, \dots, n_s\}$ let $x_i = N_0^{\alpha_i} z_i(t)$ where the α_i -s are the scaling parameters before time t . For each $i \in \{1, \dots, n_s\}$ and each $k \in \{1, \dots, n_R\}$ define

$$A_i = \frac{\log(x_i)}{\log(N_0)} \quad \text{and} \quad B_k = \frac{\log(\kappa'_k)}{\log(N_0)}.$$

Here A_i gives the copy-number scale of the i -th species at time t , while B_k is a time-independent quantity that gives the natural timescale of the k -th reaction, if all its reactant species have copy-numbers of order one. We compute the new scaling parameters by solving the following linear program

$$\begin{aligned} & \underset{\alpha_i, \beta_k}{\text{maximize}} && \psi \sum_{i=1}^{n_s} \frac{\alpha_i}{A_i} + \sum_{i=1}^{n_R} \frac{\beta_k}{B_k} \\ & \text{subject to} && 0 \leq \alpha_i \leq A_i, \beta_k \leq B_k \quad \forall i, k \\ & && \text{and} \quad \alpha_i \geq \beta_k + \alpha \cdot \nu_k \quad \forall i, k \text{ with } \xi_{ik} \neq 0 \quad . \end{aligned} \quad (14)$$

Essentially we are trying to maximize the values of the scaling parameters α_i, β_k such that the constraints (4) are satisfied and α_i approximately captures the copy-number scale of the i -th species at time t while β_k approximately captures the natural timescale of the k -th reaction. Maximizing these scaling parameters allows us to treat more reactions and species continuously, and therefore enhance the speed of our simulations. We weigh the first term in the objective function by ψ (with $\psi = 100$ as the default value) because the correct

selection of α_i -s is more crucial for the partitioning than the β_k -s. We now describe how we automatically adapt the partitioning in our method.

Algorithm 3: Adaptation

1. Solve the linear program (14) to find the α_i -s and β_k -s
2. Recompute the partitions $R_C(t), R_D(t)$ and $S_C(t), S_D(t)$ according to (13)
3. Recompute copy-number bounds (12)
4. Optional: Perform averaging (Algorithm 4, see next section)

Averaging of fast subnetworks

After the selection of the scaling parameters in the Adaptation algorithm (Algorithm 3) we try to apply the quasi-stationary assumption on fast subnetworks, if possible. The subnetworks that are suitable for this purpose can be precomputed at the start of the simulation procedure. This can be done for example, by selecting subnetworks based on conservation relationships between species, pseudo-linearity or by checking if a subnetwork is weakly reversible with deficiency zero (see Appendix A). Also, as the number of possible subnetworks grows rapidly with the number of reactions n_R in the network, one has to apply some heuristic to limit the search space of subnetworks. A straightforward choice would be to only consider subnetworks with a maximum number of reactions. While implementing the averaging procedure we go through the list of suitable subnetworks that only contain discrete reactions and we compute the timescale-separation $\Delta\zeta(R)$ as in (A1) for all such subnetworks. We select the subnetworks with $\Delta\zeta(R) \geq \Theta$ and adopt a greedy strategy of repeatedly selecting the largest subnetwork that has not been selected yet. This gives us a list of disjoint subnetworks on which the quasi-stationary assumption can be applied at the current time. Then we compute the first and second moments of the stationary distribution⁴³ for each selected subnetwork and accordingly update the rate constants of the reactions that connect the subnetwork with the surrounding network. For a description of the implementation details see Appendix A.

IV. COMBINATION WITH τ -LEAPING

In this section we propose a possible extension of this work by combining τ -leaping schemes with our method to improve its speed or accuracy. Instead of only using two regimes in our simulation, i.e. discrete stochastic jump dynamics and continuous deterministic dynamics, we can add another regime in between where the dynamics is approximated by τ -leaping. To determine which reactions are approximated by τ -leaping and which ones are approximated by continuous deterministic dynamics, the partitioning in (13) can be modified appropriately to introduce a third set of reactions R_τ that are approximated with τ -leaping. The step size τ of the τ -leap can be computed as described in¹⁴:

$$\tau = \min_{i:I} \left\{ \frac{\max \{\epsilon x_i / g_i, 1\}}{|\mu_i(x)|}, \frac{\max \{\epsilon x_i / g_i, 1\}^2}{|\sigma_i^2(x)|} \right\},$$

where

$$I = \{i : \xi_{ik} \neq 0 \text{ with } k \in R_\tau \cup R_C\}$$

is the set of species modified by the reactions in R_τ or R_C . The functions $\mu_i(x)$ and $\sigma_i^2(x)$ are given by

$$\begin{aligned} \mu_i(x) &= \sum_{k \in R_\tau \cup R_C} \xi_{ik} \lambda_k(x) \text{ and} \\ \sigma_i^2(x) &= \sum_{k \in R_\tau \cup R_C} \xi_{ik}^2 \lambda_k(x) \quad , \end{aligned}$$

where the parameters ϵ and g are usually taken to be $\epsilon = 0.03$ and $g_i = 3$. It is important to also include the continuous reactions into the step size selection as the leap condition, i.e. the change of reaction propensities during the leap has to be small, might not be satisfied otherwise. Because of our partitioning scheme there is no need to consider reactions involving low-copy number species in a special manner as is done in¹⁴ by introducing the so-called “critical reactions”. The integration of the continuous dynamics of the system can then be performed with a fixed time step integration scheme where the step size is equal to τ .

V. NUMERICAL EXAMPLES

In this section we consider three examples from Systems Biology and demonstrate that our adaptive hybrid scheme accurately captures the solution of a CME, and it can outperform

the standard SSA and fixed PDMP schemes. In one of the examples we also compare our scheme to an existing adaptive PDMP method by Alfonsi et al.²⁸.

The first example (*Fast Dimerization*) illustrates the usefulness of the averaging procedure. In this example partitioning of reactions and species is not required and our scheme reduces to the slow-scale SSA given in Cao et al.¹². However, the advantage of our method is that the quasi-stationary assumption is applied automatically without any need for explicit computations of the stationary distributions.

The second example (*Toggle Switch*) shows the performance improvements of PDMP schemes over SSA. In this example there exists a partitioning of the reactions and the species that is valid for the whole simulation period. Our method automatically chooses an appropriate partitioning, without the need for preceding manual analysis of the network, unlike other existing PDMP schemes. We also show results of the method proposed by Alfonsi et al.²⁸. Note that we implemented this method in an adaptive fashion analogous to our approach and used the same ODE solver (i.e. the Dormand-Prince 5 stepper from the boost odeint library) for the continuous dynamics.

Finally, the third example (*Repressilator*) demonstrates the advantages of using an Adaptive PDMP scheme for networks that exhibit large variations in the copy-numbers of their species. In this example a fixed partitioning will not work and hence our adaptive method will outperform any fixed PDMP approach in terms of computational efficiency and accuracy.

All simulations were run in parallel on a grid using 128 cores and using the parameter values for our algorithm given in Table II. A comparison of the runtimes for each example is given in Figure 1.

N_0	δ	μ	η	Θ
1000	1.0	1.0	0.9	0.5

TABLE II. Parameters values used for all examples.

A. Fast Dimerization

The fast dimerization SRN¹² consists of $n_S = 3$ species s_0 , s_1 and s_2 and $n_R = 4$ reactions and is depicted in Figure 2. The reactions are listed in Table III.

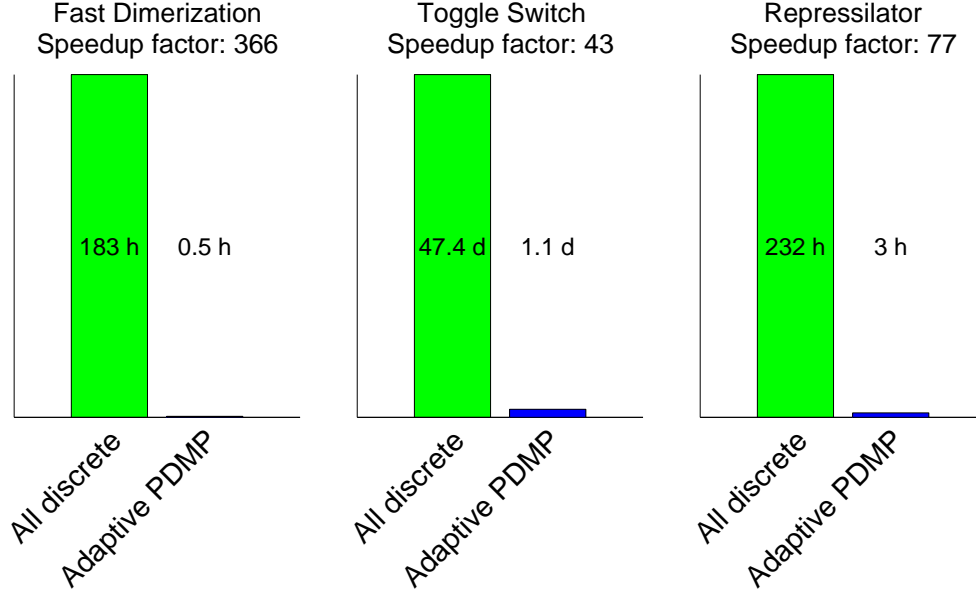


FIG. 1. Runtime comparison of fully discrete dynamics and our Adaptive PDMP method. Shown is the comparison of total CPU time required for 100'000 sample paths.

Reaction	κ'	Reaction	κ'
$s_0 + s_0 \rightarrow s_1$	1.0	$s_1 \rightarrow s_0 + s_0$	200.0
$s_0 \rightarrow \emptyset$	0.02	$s_1 \rightarrow s_2$	0.004

TABLE III. Reactions of the Fast Dimerization network

The simulation is run from $t_0 = 0$ until $t_f = 400$ and the initial state is set to $x(t_0) = x_0 = (540, 730, 0)$. Considering the difference in the orders of magnitude of the rate constants for reaction $s_0 + s_0 \rightarrow s_1$ and $s_1 \rightarrow s_0 + s_0$ compared to the rate constants for reaction $s_0 \rightarrow \emptyset$ and $s_1 \rightarrow s_2$, it seems plausible that the dynamics of a subnetwork consisting of the species s_0 and s_1 can be averaged. Indeed our adaptive scheme identifies the subnetwork consisting of species s_0 and s_1 as suitable for averaging, and using the strategy for zero-deficiency networks the quasi-stationary assumption can be applied. This can be clearly seen in Figure 3, where the copy-numbers of species s_0 and s_1 show strong fluctuations for the SSA simulations and only weak fluctuations for the Adaptive PDMP simulations (the weak fluctuations are due to the reaction $s_1 \rightarrow s_2$ which is still simulated with discrete dynamics). The averaging provides a huge performance gain without losing any essential information about the dynamics of the species s_2 .

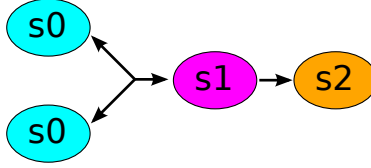


FIG. 2. Cartoon of the Fast Dimerization network

The simulation of 100'000 sample paths with SSA took a total CPU time of ≈ 183 hours. The simulation of 100'000 sample paths with our Adaptive PDMP scheme using zero-deficiency averaging took a total CPU time of ≈ 0.5 hours. Figure 3 shows the results of these simulations. The distribution given by the CME (estimated with SSA) closely matches the distribution estimated with our Adaptive PDMP scheme at time $t = 200$ (see Figure 4). The Kolmogorov-Smirnov distance between the two distributions is $d_{KS} = 0.01$.

1. Comparison to an existing adaptive PDMP method

We compare our Adaptive PDMP scheme with an existing adaptive PDMP method proposed by Alfonsi et al.²⁸. We implemented the method similarly to our own implementation and added the necessary details. We compare both methods with SSA for the Fast Dimerization SRN without averaging.

Figure 5 shows the results of 10'000 sample paths for each method. One can clearly see a good match of our method with SSA. The results from the method by Alfonsi et al. are inaccurate. In this case, we suspect that this is due to the partitioning criteria that only considers the copy-number of reactants. Because of this the reaction $s_1 \rightarrow s_2$ is considered as being continuous when the copy-number of s_1 is big enough. Thus the stochastic variation in species s_2 is strongly reduced in comparison to the exact SSA simulations.

B. Toggle Switch

We consider the Toggle-Switch network, which is similar to the synthetic Toggle-Switch by Gardner et al.⁵, but implemented with mass-action kinetics. It consists of $n_S = 6$ species $m_A, s_A, m_B, s_B, p_A, p_B$ and $n_R = 16$ reactions and is depicted in Figure 6. The reactions are listed in Table IV.

We estimate the solution given by the CME at time $t_f = 10^5$ where the initial state is set

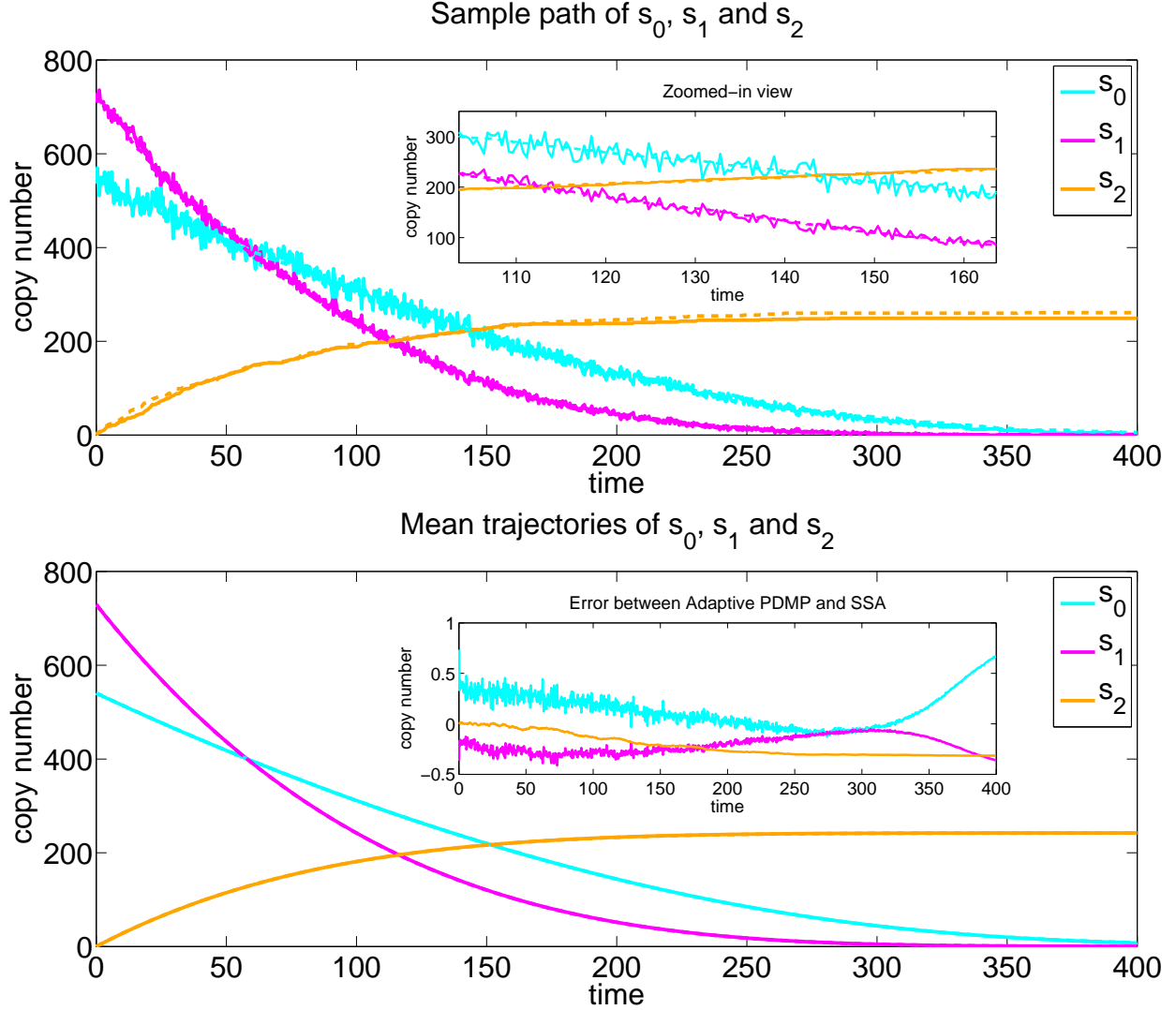


FIG. 3. Simulation results of the Fast Dimerization SRN

The upper plot shows a sample path of species s_0 , s_1 and s_2 with SSA (solid lines) and with our Adaptive PDMP scheme (dashed lines). The lower plot shows the mean copy-number of species s_0 , s_1 and s_2 of 100'000 simulations with SSA and our Adaptive PDMP scheme.

to $x_0 = (0, \dots, 0)$. As our scheme could not find any suitable subnetworks for this example, we turned off the averaging procedure to reduce the computational overhead. There are two high copy-number proteins, p_A and p_B , which are controlled by two low copy-number mRNAs, s_A and s_B . These mRNAs s_A and s_B are the processed variants of the precursor mRNAs m_A and m_B . In addition to inducing translation the mRNAs, s_A and s_B also induce the degradation of m_A and m_B respectively. The network shows a bistable behaviour and

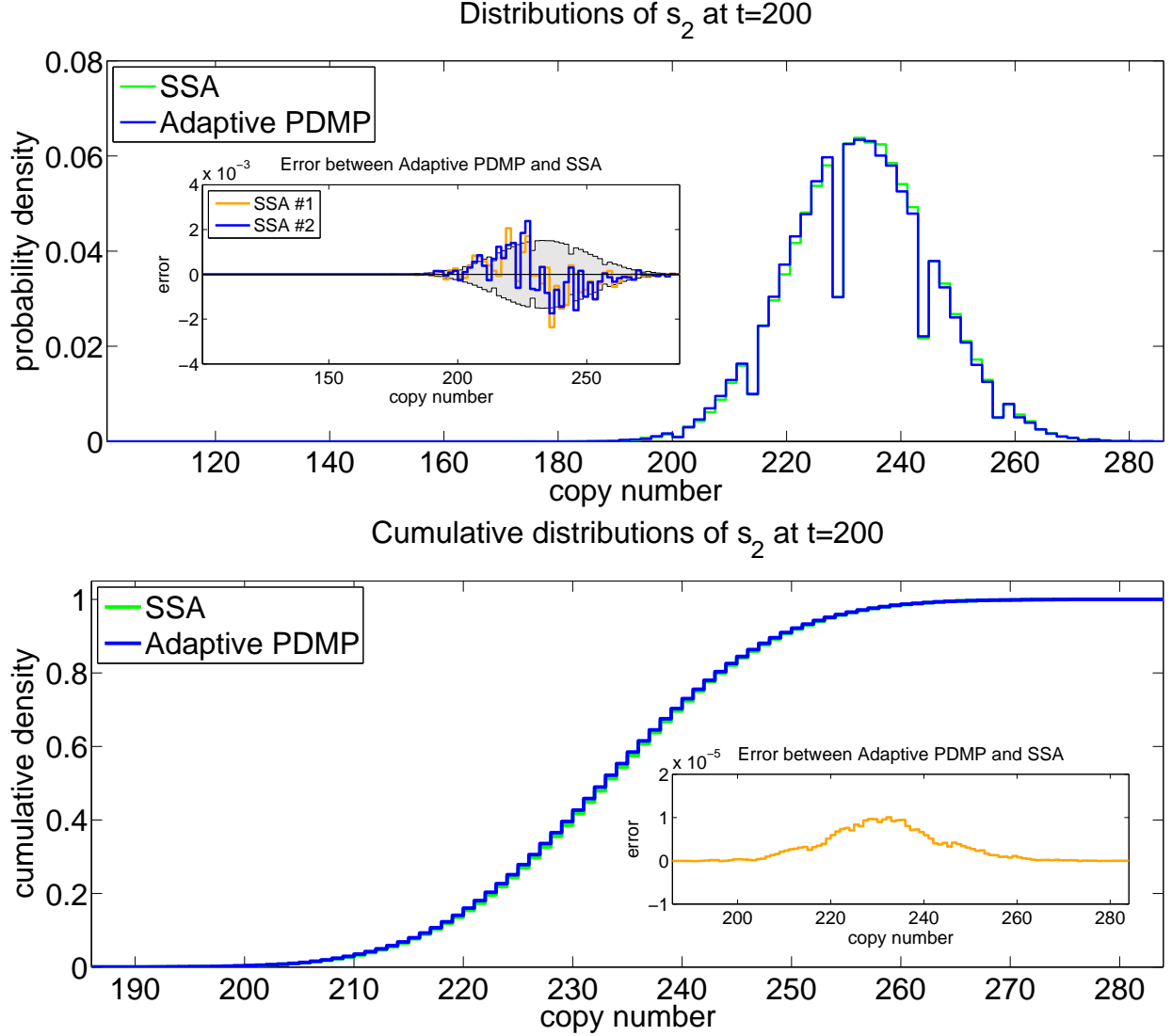


FIG. 4. Probability distribution of the Fast Dimerization SRN

The upper plot shows the copy-number distribution (100 equally spaced bins) of species s_2 at time $t = 200$ with SSA (green) and with our Adaptive PDMP scheme (blue). The inset shows the error compared to two independent runs with SSA, each with 100'000 samples. The grey shaded area marks the 95% confidence interval of the SSA run #1.

The lower plot shows the cumulative copy-number distributions of species s_2 at time $t = 200$ with SSA (green) and with our Adaptive PDMP scheme (blue).

can stochastically switch between a high copy-number p_A , low copy-number p_B state and a low copy-number p_A , high copy-number p_B state. While the SSA will spend a major part of the simulation time for simulating the translation and degradation of the p_A and p_B proteins,

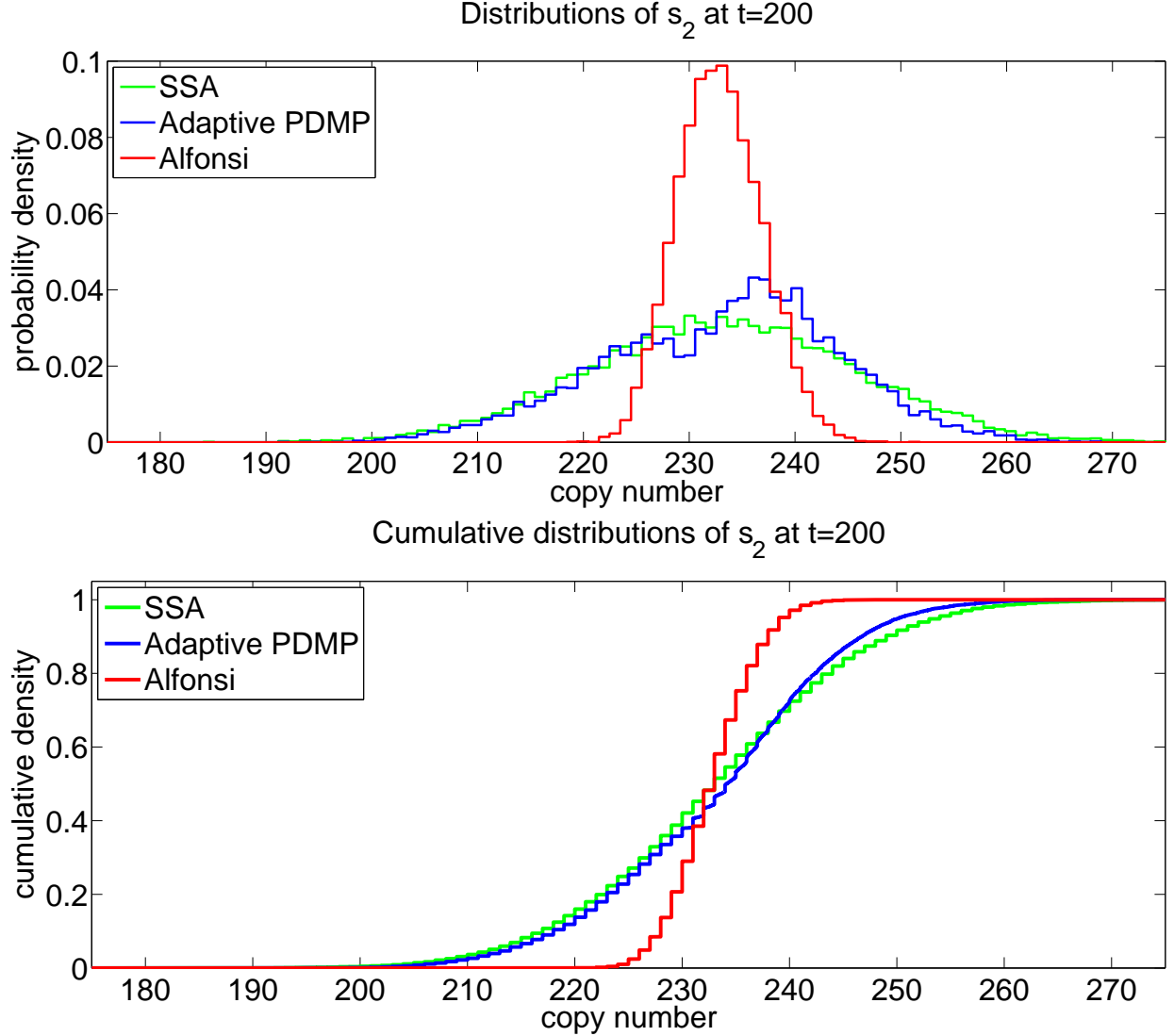


FIG. 5. Comparison of the probability distribution of the Fast Dimerization SRN

The upper plot shows the copy-number distribution (100 equally spaced bins) of species s_2 at time $t = 200$ with SSA (green), with our Adaptive PDMP (blue) and with the Alfonsi scheme (red). The lower plot shows the cumulative copy-number distributions of species s_2 at time $t = 200$ with SSA (green), with our Adaptive PDMP (blue) and with the Alfonsi scheme (red).

our adaptive PDMP scheme performs those reactions with continuous dynamics when the proteins have a high copy-number, yielding a considerable performance gain.

The simulation of 100'000 sample paths with SSA took a total CPU time of ≈ 47.4 days. The simulation of 100'000 sample paths with our Adaptive PDMP scheme using a step size of $dt = 5.0$ took a total CPU time of ≈ 1.1 days. Figure 7 shows the results of these simulations.

Reaction	κ'	Reaction	κ'	Reaction	κ'
$\emptyset \rightarrow m_A$	1.0	$m_A \rightarrow s_A$	5.0	$s_A \rightarrow \emptyset$	0.01
$\emptyset \rightarrow m_B$	1.0	$m_B \rightarrow s_B$	5.0	$s_B \rightarrow \emptyset$	0.01
$m_A \rightarrow \emptyset$	0.1	$s_A + m_B \rightarrow s_A$	20.0	$s_A \rightarrow s_A + p_A$	10.0
$m_B \rightarrow \emptyset$	0.1	$s_B + m_A \rightarrow s_B$	20.0	$s_B \rightarrow s_B + p_B$	10.0
$p_A \rightarrow \emptyset$	0.1	$p_B \rightarrow \emptyset$	0.1		

TABLE IV. Reactions of the Toggle Switch network

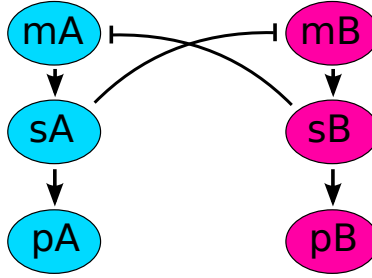


FIG. 6. Cartoon of the Toggleswitch network

The distribution given by the CME (estimated with SSA) closely matches the distribution estimated with our Adaptive PDMP scheme at time $t = 25'000$. The Kolmogorov-Smirnov distance between the two distributions is $d_{KS} = 0.01$.

C. Repressilator

We consider the Repressilator network, which is similar to the synthetic Repressilator by Elowitz et al.⁶, but implemented with mass-action kinetics. It consists of $n_S = 6$ species $m_A, p_A, m_B, p_B, m_C, p_C$ and $n_R = 15$ reactions and is depicted in Figure 8. The reactions are listed in Table V.

We estimate the solution given by the CME at time $t_f = 5 \times 10^4$ where the initial state is set to $x_0 = (10, 500, 0, 0, 0, 0)$. Also in this example, there were no suitable subnetworks for the averaging procedure, so we reduced the computational overhead by turning it off.

The Repressilator consists of three mRNAs m_A, m_B and m_C and the three corresponding proteins p_A, p_B and p_C . Each protein catalyses the degradation of another mRNA in a

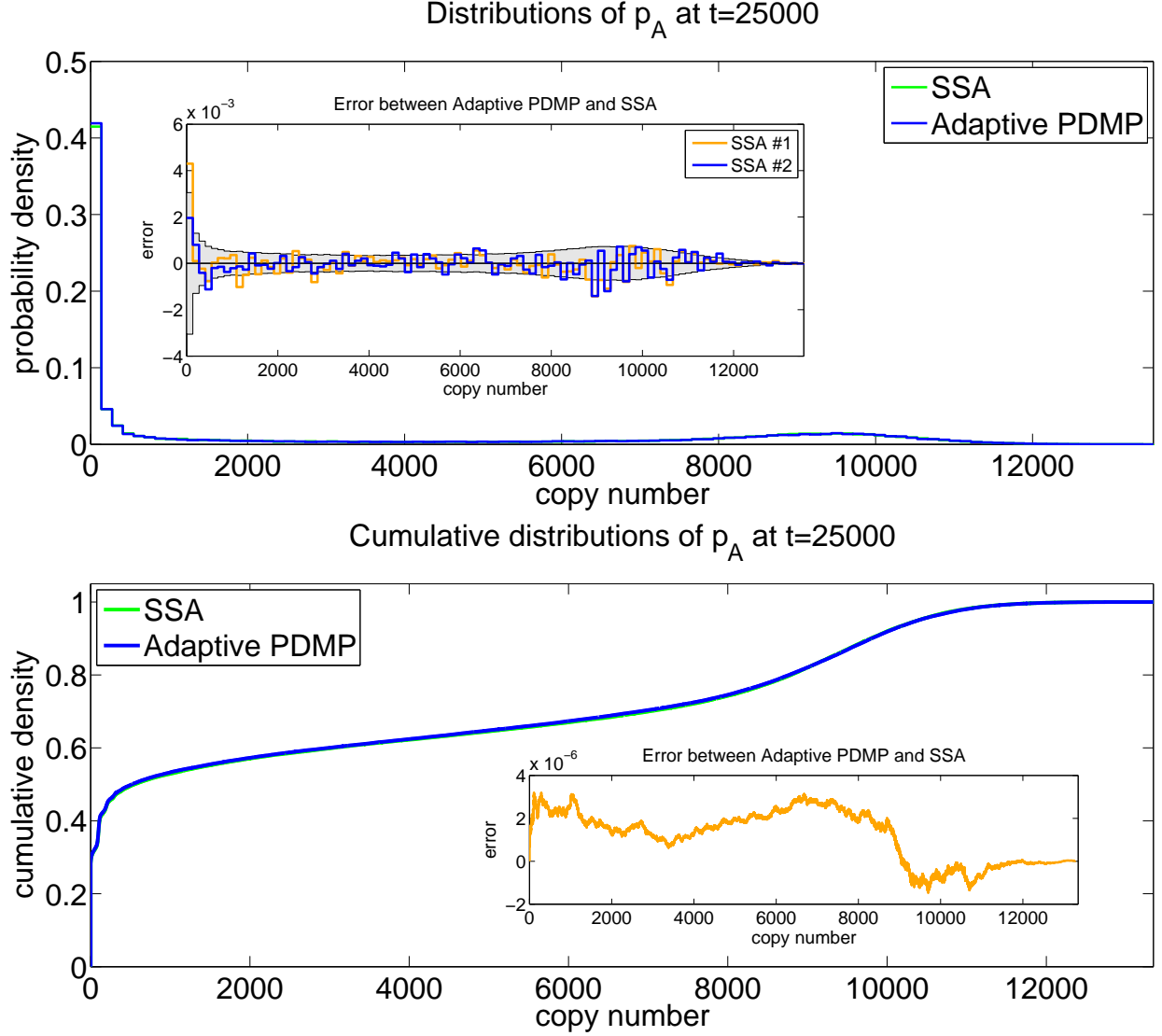


FIG. 7. Probability distribution of the Toggleswitch SRN

The upper plot shows the copy-number distribution (100 equally spaced bins) of species p_A at time $t = 25'000$ with SSA (green) and with our Adaptive PDMP scheme (blue). The inset shows the error compared to two independent runs with SSA, each with 100'000 samples. The grey shaded area marks the 95% confidence interval of the SSA run #1. The lower plot shows the cumulative copy-number distributions of species p_A at time $t = 25'000$ with SSA (green) and with our Adaptive PDMP scheme (blue).

circular manner, that is, p_A degrades m_C , p_B degrades m_A and p_C degrades m_B . In the stochastic setting, this network exhibits spontaneous oscillatory behaviour (see Figure 9).

Reaction	κ'	Reaction	κ'	Reaction	κ'
$\emptyset \rightarrow m_A$	0.1	$\emptyset \rightarrow m_B$	0.1	$\emptyset \rightarrow m_C$	0.1
$m_A \rightarrow m_A + p_A$	50.0	$m_B \rightarrow m_B + p_B$	50.0	$m_C \rightarrow m_C + p_C$	50.0
$m_A \rightarrow \emptyset$	0.01	$m_B \rightarrow \emptyset$	0.01	$m_C \rightarrow \emptyset$	0.01
$m_A + p_B \rightarrow p_B$	50.0	$m_B + p_C \rightarrow p_C$	50.0	$m_C + p_A \rightarrow p_A$	50.0
$p_A \rightarrow \emptyset$	0.01	$p_B \rightarrow \emptyset$	0.01	$p_C \rightarrow \emptyset$	0.01

TABLE V. Reactions of the Repressilator network

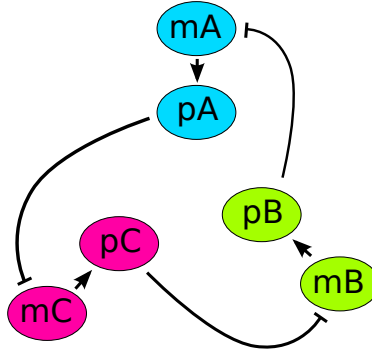


FIG. 8. Cartoon of the Repressilator network

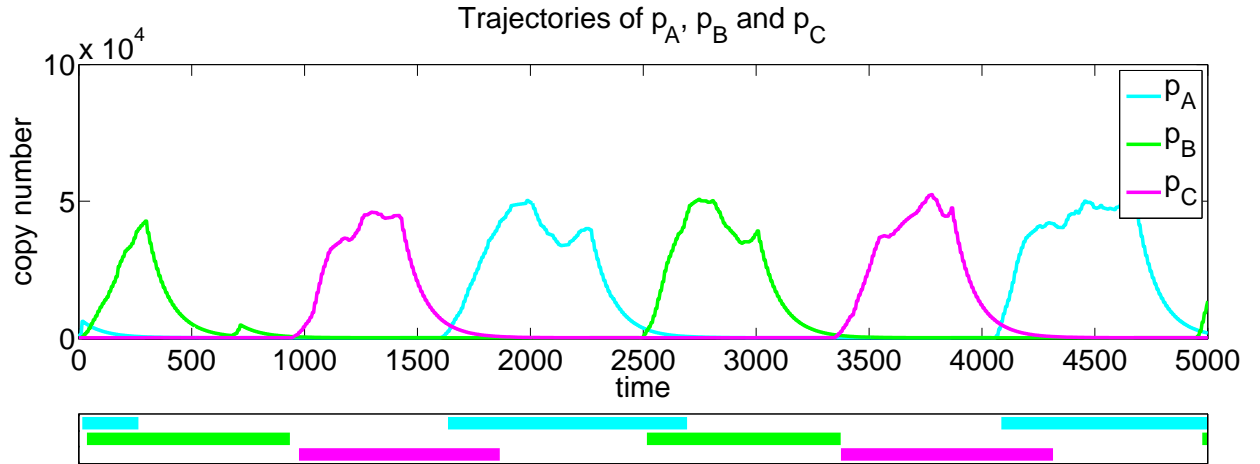


FIG. 9. Single trajectory showing the adaptation of our method

Shown is a single sample-path of the copy-number for species p_A , p_B and p_C simulated with our Adaptive PDMP scheme. The colored areas below indicate when the corresponding species p_X and the reactions $m_X \rightarrow m_X + p_X$ and $p_X \rightarrow \emptyset$ are considered to be continuous.

A fixed PDMP scheme that simulates reactions affecting only the protein species continuously (i.e. $p_X \rightarrow \emptyset$ and $m_X \rightarrow m_X + p_X$) and the rest discretely, does not exhibit the oscillatory behaviour that can be seen by simulating the SRN with SSA. This can be seen in Figure 10 where the first moment of the copy-number of species p_A is shown for different simulation schemes. The result of a fixed PDMP scheme that simulates the translation- and degradation-reaction of species p_A with continuous dynamics is shown in red and shows a quick drop of the copy-number to 0. This is due to the discrete nature of the species in the stochastic description. In the fixed PDMP scheme the protein p_A with a very small concentration can still degrade the mRNA m_C , whereas in the stochastic setting the copy-number of protein p_A can drop to 0 and will not contribute to the degradation of m_C anymore, thus enabling m_C to rise. On the other hand, when the copy-numbers of the proteins are high, the stochasticity of the translation- and the degradation-reaction is negligible.

Our Adaptive PDMP scheme can approximate the protein dynamics continuously when they have high copy-numbers and switch to discrete stochastic simulations when they have low copy-numbers. This is depicted in Figure 9 where the colored bars indicate when the corresponding degradation- and translation-reactions are approximated with continuous dynamics. One can easily see that the dynamics for the degradation- and translation-reaction of p_B is approximated with continuous dynamics when the copy-number of p_B is high. In Figure 10 of simulations with SSA (green) and with our adaptive method (blue) show the same oscillatory behaviour. Due to stochasticity the average amplitude decreases with time as the individual trajectories lose their synchronization, unlike the deterministic setting. Note that the initial state is the same for each trajectory. Hence this example illustrates, how an adaptive partitioning scheme can be very useful.

The simulation of 100'000 sample paths with SSA took a total CPU time of ≈ 232 hours. The simulation of 100'000 sample paths with our Adaptive PDMP scheme using a step size of $dt = 1.0$ took a total CPU time of ≈ 3 hours. Figure 11 shows the distribution of species p_A at time $t = 4'750$ for simulations with SSA (green) and with our adaptive scheme (blue). In addition the insets show the deviation from two independent runs with SSA and also mark the 95% confidence intervals (grey area). The distribution given by the CME (estimated with SSA) closely matches the distribution estimated with our Adaptive PDMP scheme at time $t = 4'750$. The Kolmogorov-Smirnov distance between the two distributions is $d_{KS} = 0.003$.

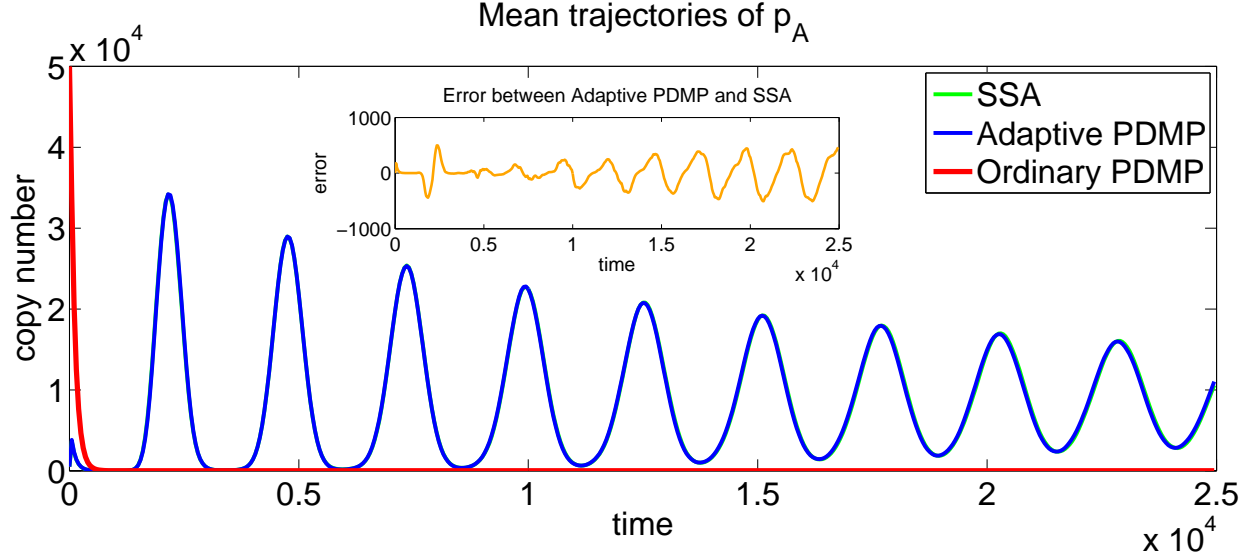


FIG. 10. Mean trajectories of species p_A for the Repressilator SRN

Shown is the mean of species p_A of 100'000 simulations with SSA (green), with the fixed PDMP scheme (red) and with our Adaptive PDMP scheme (blue).

VI. CONCLUSIONS AND FUTURE WORK

In this paper we propose a novel adaptive hybrid scheme for approximating the solution of a Chemical Master Equation for Stochastic Reaction Networks spanning a wide range of reaction timescales, and having large variation in species copy-number scales. The method is based on a rigorous mathematical framework by Kang et al.¹⁹, that ensures the validity of the approximations.

To achieve considerable speed-ups over SSA our method introduces two sources of error: we treat high copy-number species as continuous and we assume stationarity for the dynamics of fast subnetworks. However with the help of parameters we can tune the amount of error one is willing to tolerate in a qualitative manner. For example, setting $N_0 \approx \infty$ will reduce our scheme to SSA and setting $\Theta \approx \infty$ would prevent the application of the quasi-stationary assumption.

We compare the results of our adaptive method with SSA and a fixed PDMP scheme using examples from Systems Biology. Our results show that the adaptive method can provide considerable performance enhancements in comparison to both these approaches. Moreover, as our last example demonstrates, the adaptivity can be crucial to achieve accuracy and

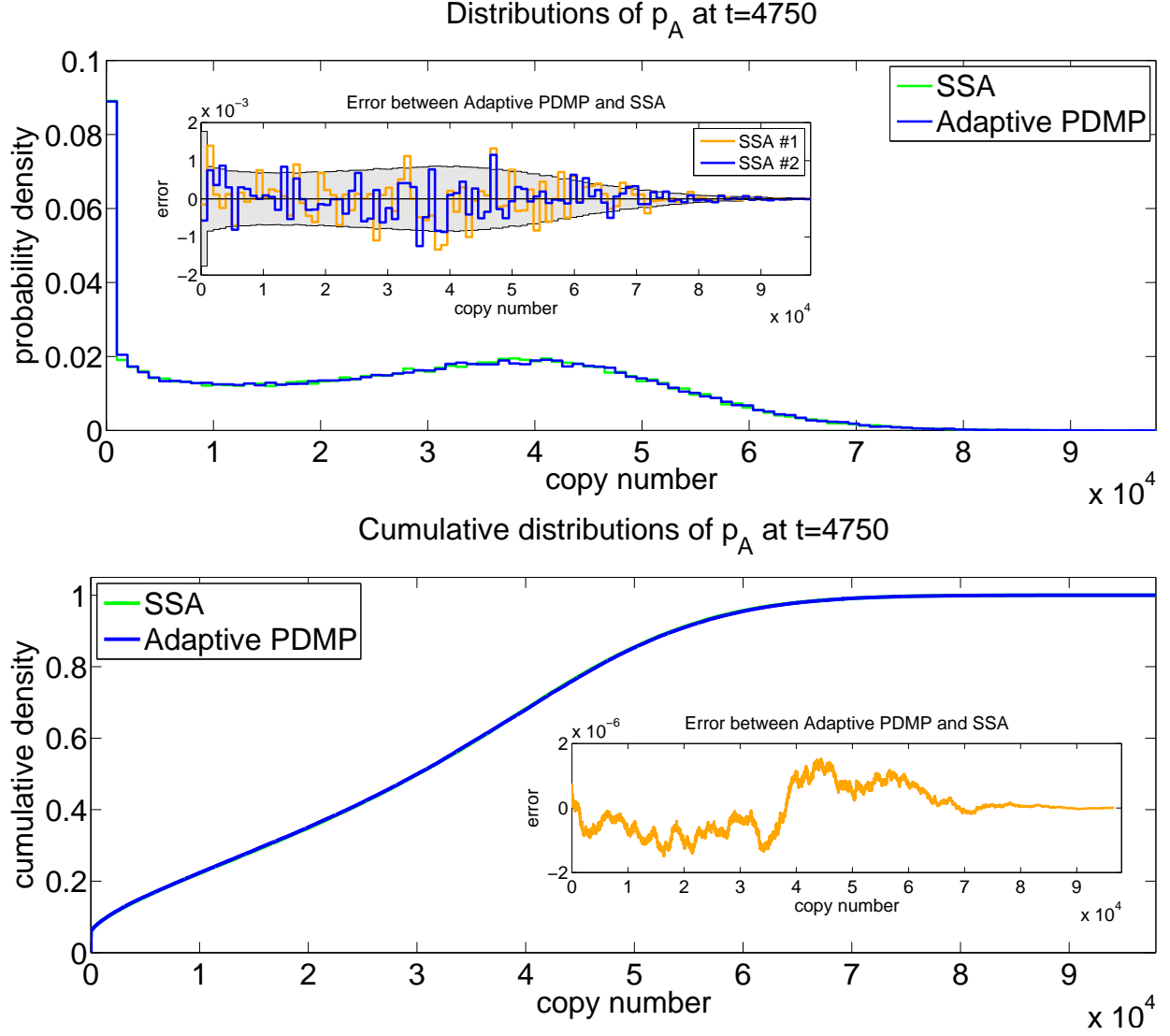


FIG. 11. Simulation results of the Repressilator SRN

The upper plot shows the copy-number distribution (100 equally spaced bins) of species p_A at time $t = 4750$ with SSA (green) and with our Adaptive PDMP scheme (blue). The inset shows the error compared to two independent runs with SSA, each with 100'000 samples. The grey shaded area marks the 95% confidence interval of the SSA run #1.

The lower plot shows the cumulative copy-number distributions of species p_A at time $t = 4750$ with SSA (green) and with our Adaptive PDMP scheme (blue).

reduce the computational cost at the same time.

The adaptive feature of our method makes it particularly well suited for SRNs where the timescales of the reaction dynamics can change with time. Such behaviour can be expected

in regulatory gene expression networks. We hope that our method will provide a tool for researchers to analyse more complicated and realistic models in Systems Biology.

Our adaptive scheme can potentially be used in the simulation of large scale compartment models of Spatial SRNs⁴⁴. For such systems the spatial distribution of species can change with time and our adaptive method could reduce the computational costs significantly. For example, in the Min System in *E. coli*⁴, the location of high copy-number species is constantly changing. So an adaptive approach could neglect the fluctuations at these locations, while conserving the fluctuations at locations with low copy-numbers. In the future we aim to develop such adaptive schemes for Spatial SRNs.

Another direction for further research is a combination of our method with a τ -leaping scheme, as we propose in Section IV, and the corresponding study of the mathematical validity of such an approach.

Appendix A: Averaging of fast subnetworks

We now describe the averaging procedure for fast subnetworks in this section which is an optional extension to the previously discussed approximation of SRNs.

To check if a subnetwork has *fast* dynamics in comparison to the surrounding network (which consists of species that are directly influenced by the subnetwork) we look at two sets of reactions, the reactions within the subnetwork, and the reactions connecting the subnetwork to the surrounding network. We then define a timescale for both these sets of reactions and describe a formal criterion to check whether the subnetwork is fast.

As an example, Figure 12 shows a network where a transcription factor switches between an active (tf*) and an inactive (tf) form. In the active form it can bind to a gene and initiate transcription. Assuming that the switch between the active and inactive form of the transcription factor is fast in comparison to the binding of the active transcription factor to the gene, the red reaction would define a fast subnetwork. The orange species make up the fast subnetwork and the blue species make up the surrounding network. The green reaction connects the subnetwork to the surrounding network.

Now we describe this procedure elaborately. Consider a subnetwork with m_R reactions $R \subseteq \{1, \dots, n_R\}$. We define the set of species involved in the reactions in R as

$$Q(R) = \{i : \xi_{ik} \neq 0 \text{ for some } k \in R\}$$

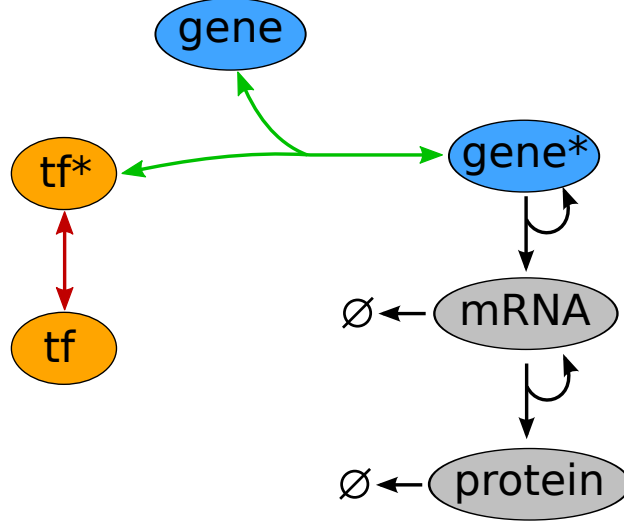


FIG. 12. Example network depicting a fast subnetwork and the corresponding reactions

and define the set of species that are catalytically involved in the reactions in R as

$$Q_C(R) = \{i \notin Q(R) : \nu_{ik} = \nu'_{ik} > 0 \text{ for some } k \in R\}.$$

Note that the propensities of the reactions in R only depend on the species in $Q(R)$ and $Q_C(R)$. Assume that the copy-numbers of the species in $Q_C(R)$ are constant. Then the reactions in R and the species in $Q(R)$ naturally define a SRN. Suppose that the dynamics of this SRN converges to a unique stationary distribution for any choice of copy-numbers of the species in $Q_C(R)$. If the reactions in R are sufficiently faster than the surrounding reactions, given by

$$R_S(R) = \{k \notin R : \nu_{ik} > 0 \text{ or } \nu'_{ik} > 0 \text{ for some } i \in Q(R)\} \\ \cup \{k \notin R : \xi_{ik} \neq 0 \text{ for some } i \in Q_C(R)\},$$

then the stationary distribution of the subnetwork can be used to compute the propensities of the reactions in $R_S(R)$. This is called the quasi-stationary assumption¹².

We now specify a formal criteria to check if the subnetwork dynamics is sufficiently fast in comparison to its surrounding network. For this we define a timescale for each reaction k as

$$\zeta(k) = \frac{\log \kappa'_k}{\log N_0} + \sum_{i=1}^{n_S} \nu_{ik} \cdot \frac{\log x_i}{\log N_0} \cdot \mathbb{1}_{\{x_i > 0\}}$$

and the observation timescale as $\zeta(0) = \frac{\log(1/t_f)}{\log N_0}$. The form of $\zeta(k)$ is chosen in such a way that $N_0^{\zeta(k)}$ captures the timescale of the k -th reaction channel. However, when reaction

channel k consumes a species i (i.e. $\xi_{ik} < 0$) and the copy-number $x_i = 0$, then the exponent to N_0 would have to be $-\infty$ to exactly capture the timescale of the k -th reaction channel. A single reaction producing species i would then change the timescale to a finite value and immediately render the averaging decision based on the timescale as invalid. To prevent this we overestimate the timescale of a reaction channel by ignoring the $-\infty$ terms when computing $\zeta(k)$. This is achieved by adding the indicator functions $\mathbb{1}_{\{x_i > 0\}}$ in $\zeta(k)$.

The timescale separation of the fast subnetwork is then given by

$$\Delta\zeta(R) = \min_{k \in R} (\zeta(k)) - \max_{k \in R_S(R) \cup \{0\}} (\zeta(k)) \quad . \quad (\text{A1})$$

In words, $\Delta\zeta(R)$ is the difference between the slowest timescale of all reactions within the subnetwork and the fastest timescale of all reactions connecting the subnetwork to the surrounding network. We fix a positive parameter Θ (with $\Theta = 0.5$ as the default value). If $\Delta\zeta(R) \geq \Theta$ then we call the subnetwork defined by R and $Q(R)$ to be *fast*.

Given that a subnetwork is fast, we can apply the quasi-stationary assumption if the dynamics of this subnetwork converges to a stationary distribution. Below we describe three different strategies to check if this is indeed the case.

Finite Markov Chains

Suppose that the species in $Q(R)$ satisfy a conservation relation of the form $\sum_{i \in Q(R)} b_i x_i = C$, where the b_i -s and C are positive integers and assume that the copy-numbers of species outside $Q(R)$ are fixed. Due to the reactions in R the copy-number vector of all the species is constrained to reside in a finite set, which can be enumerated as $\{e_1, \dots, e_L\}$. Define a $L \times L$ matrix A with entries given by

$$A_{ij} = \begin{cases} \sum_{R_{ij}} \lambda'_k(e_i) & \text{if } i \neq j \\ -\sum_{k \in R} \lambda'_k(e_i) & \text{if } i = j \end{cases}$$

where $R_{ij} = \{k \in R : e_i + \xi_k = e_j\}$. If there exists a unique positive vector $\pi = (\pi_1, \dots, \pi_L)$ such that $\sum_{i=1}^L \pi_i = 1$ and $\pi A = 0$, then the Markov process corresponding to the subnetwork dynamics is ergodic with the unique stationary distribution given by the vector π . One can check the existence and uniqueness of π by verifying that the dimension of the left null-space of A is one⁴⁵.

Pseudo-Linear subnetworks

Consider the subnetwork defined by R and $Q(R)$ as before. Assume that the propensities of the reactions in R are affine functions of the copy-numbers of the species in $Q(R)$, that is $\sum_{i \in Q(R)} \nu_{ik} \leq 1$ for all $k \in R$. Such a subnetwork is called *pseudo-linear* because it does not involve any bimolecular reactions between the fast species in the subnetwork. For linear networks the moment equations are closed. Hence we can easily find the first two stationary moments by finding the fixed points of the moment equations

$$\begin{aligned} \frac{d\mathbb{E}[X_i]}{dt} &= \sum_{k \in R} \xi_{ik} \mathbb{E}[\lambda'_k(X)] \\ \frac{d\mathbb{E}[X_i X_j]}{dt} &= \sum_{k \in R} \xi_{ik} \mathbb{E}[\lambda'_k(X) X_j] + \sum_{k \in R} \mathbb{E}[\lambda'_k(X) X_i] \xi_{jk} + \sum_{k \in R} \xi_{ik} \mathbb{E}[\lambda'_k(X)] \xi_{jk}, \end{aligned}$$

and checking if these fixed points are stable. As we only consider at most bimolecular reactions we only need the first two stationary moments to apply the quasi-stationary assumption.

Zero-Deficiency subnetworks

In some situations, a fast subnetwork may not be pseudo-linear and its state space is either infinite or too large to be able to compute the exact stationary distribution for the corresponding Markov Chain (for example see the *Fast Dimerization* network in Section V). In such cases it is sometimes possible to use the recent results from Anderson et al.⁴⁶ to compute the stationary distribution of the subnetwork dynamics. In particular Anderson et al.⁴⁶ show that *weakly reversible*⁴⁷ networks with zero *deficiency*⁴⁸ admit a product-form stationary distribution given by

$$\pi(x) = M \prod_{i \in Q(R)} \frac{c_i^{x_i}}{x_i!}, \quad x \in \Gamma \tag{A2}$$

and $\pi(x) = 0$ otherwise, where Γ is an irreducible state space containing the initial state, $c \in \mathbb{R}_{\geq 0}^{n_s}$ is the equilibrium point of the corresponding deterministic system and M is a positive normalizing constant. We can find the appropriate set Γ using the results in⁴⁹, and this would imply that the dynamics of the distribution of the SRN will converge to the stationary distribution π .

Implementation details

During the simulation of the sample path we need to keep track of changes to the copy-numbers of species in $Q(R)$, so that we can recompute the moments of the stationary distribution of the fast subnetwork. We do this by modifying the partitioning of the reactions, so that the reactions in $R_S(R)$ are discrete. Every time a reaction in $k \in R_S(R)$ occurs, the first and second moments of the stationary distribution are recomputed and the appropriate rate constants are updated. For this, Algorithm 2 has to be modified in a straightforward manner. Below we describe the averaging procedure.

Algorithm 4: Averaging

1. Once: Precompute subnetworks L_A suitable for averaging
(e.g. weakly reversible, zero-deficiency subnetworks)
2. Let L_P be the set of previously averaged subnetworks
3. Set $L_C = \emptyset$
4. for each suitable subnetwork $R \in L_A$ do
5. if $\Delta\zeta(R) \geq \Theta$ then
6. Set $L_C = L_C \cup \{R\}$
7. end if
8. end for
9. $L_F = \emptyset$
10. while $L_C \neq \emptyset$ do
11. Set $R = \operatorname{argmax}_{W \in L_C} (|W|)$
12. Set $L_C = L_C \setminus R$
13. if $\bigcup_{W \in L_F} W \cap R = \emptyset$ and
 $\bigcup_{W \in L_F} Q(W) \cap Q(R) = \emptyset$ then
14. Set $L_F = L_F \cup \{R\}$
15. Modify R_D, R_C , so that $R_S(R) \subseteq R_D$
16. end if
17. end while
18. for each subnetwork $R \in L_P \setminus L_F$ do
19. Sample state of the species in $Q(R)$ from the

- stationary distribution
- 20. end for
- 21. for each subnetwork $R \in L_F$ do
- 22. Compute stationary first and second moment of the
species in $Q(R)$
- 23. Update the stoichiometries and propensities of
reactions in $R_S(R)$
- 24. end for

Appendix B: Mathematical Justification

In this section we provide a mathematical justification of why our adaptive scheme produces a close approximation to the solution of a CME. We start with a simple lemma that shows convergence of a sequence of processes, in which each process is formed by stitching together two Markov processes at a random stopping time determined by the first process. In this section let $n = n_S$, where n_S is the number of species, and let “ \Rightarrow ” and “ \approx_d ” denote convergence in distribution and similarity in distribution respectively. Let $\mathbb{D}_{\mathbb{R}^n}[0, \infty)$ denote the space of càdlàg functions, from $[0, \infty)$ to \mathbb{R}^n , that is, right continuous functions with left limits³⁹. Let A be a measurable subset of \mathbb{R}^n and define $\mathcal{G}_A : \mathbb{D}_{\mathbb{R}^n}[0, \infty) \rightarrow [0, \infty]$ by

$$\mathcal{G}_A(\zeta) = \inf \{t \geq 0 : \zeta(t) \notin A\}. \quad (\text{B1})$$

For each N , let W_1^N, W_2^N be two Markov processes with sample paths in $\mathbb{D}_{\mathbb{R}^n}[0, \infty)$. Assume that for any sequence of initial conditions satisfying $W_i^N(0) \Rightarrow W_i(0)$ we have $W_i^N \Rightarrow W_i$ as $N \rightarrow \infty$, where W_i is a PDMP as in (5). Define a stopping time $\tau^N = \mathcal{G}_A(W_1^N)$. Suppose $W_1^N(0) = w_0$ and $W_2^N(0) = f(W_1^N(\tau^N))$, where $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a measurable function.

Define another process W^N by

$$W^N(t) = W_1^N(t) \mathbb{1}_{\{t < \tau^N\}} + W_2^N(t - \tau^N) \mathbb{1}_{\{t \geq \tau^N\}}$$

Lemma. *For any $t \geq 0$ we have*

$$W^N(t) \Rightarrow W(t)$$

where

$$W(t) = W_1(t) \mathbb{1}_{\{t < \tau\}} + W_2(t - \tau) \mathbb{1}_{\{t \geq \tau\}}$$

with $\tau = \mathcal{G}_A(W_1)$, $W_1(0) = w_0$ and $W_2(0) = f(W_1(\tau))$.

Proof. Let $g : \mathbb{R}^n \rightarrow \mathbb{R}^n$ be any bounded continuous function. To prove the lemma it suffices to show that for any $t \geq 0$

$$\lim_{N \rightarrow \infty} \mathbb{E}(g(W^N(t))) = \mathbb{E}(g(W(t))), \quad (\text{B2})$$

where \mathbb{E} denotes the expectation operator. Fix a $t \geq 0$. We can write

$$\begin{aligned} \mathbb{E}(g(W^N(t))) &= \mathbb{E}(g(W^N(t))\mathbb{1}_{\{t < \tau^N\}} + g(W^N(t))\mathbb{1}_{\{t \geq \tau^N\}}) \\ &= \mathbb{E}(g(W_1^N(t))\mathbb{1}_{\{t < \tau^N\}} + g(W_2^N(t - \tau^N))\mathbb{1}_{\{t \geq \tau^N\}}). \end{aligned} \quad (\text{B3})$$

Note that $\tau^N = \mathcal{G}_A(W_1^N)$ and $W_1^N \Rightarrow W_1$, where W_1 is a PDMP. Hence, we can conclude that $(W_1^N, \tau^N) \Rightarrow (W_1, \tau)$, where $\tau = \mathcal{G}_A(W_1)$. This shows that

$$\lim_{N \rightarrow \infty} \mathbb{E}(g(W_1^N(t))\mathbb{1}_{\{t < \tau^N\}}) = \mathbb{E}(g(W_1(t))\mathbb{1}_{\{t < \tau\}}). \quad (\text{B4})$$

The second term in (B3) can be expressed as

$$\mathbb{E}(g(W_2^N(t - \tau^N))\mathbb{1}_{\{t \geq \tau^N\}}) = \mathbb{E}(\mathbb{E}(g(W_2^N(t - \tau^N))\mathbb{1}_{\{t \geq \tau^N\}} | W_1^N(\tau^N)))$$

Given $W_1^N(\tau^N) = w$, the process W_2^N converges in distribution to the PDMP W_2 with initial state $f(w)$. Moreover, the results in⁵⁰ show that $W_1^N(\tau^N) \Rightarrow W_1(\tau)$. Therefore

$$\begin{aligned} \lim_{N \rightarrow \infty} \mathbb{E}(g(W_2^N(t - \tau^N))\mathbb{1}_{\{t \geq \tau^N\}}) &= \mathbb{E}(\mathbb{E}(g(W_2(t - \tau))\mathbb{1}_{\{t \geq \tau\}} | W_1(\tau))) \\ &= \mathbb{E}(g(W_2(t - \tau))\mathbb{1}_{\{t \geq \tau\}}). \end{aligned}$$

Using this relation along with (B4) and (B3) we obtain

$$\lim_{N \rightarrow \infty} \mathbb{E}(g(W^N(t))) = \mathbb{E}(g(W_1(t))\mathbb{1}_{\{t < \tau\}}) + \mathbb{E}(g(W_2(t - \tau))\mathbb{1}_{\{t \geq \tau\}}) = \mathbb{E}(g(W(t))).$$

This shows (B2) and finishes the proof of this lemma. \square

The above lemma can be easily generalized to the more general case, where W^N is defined by stitching together m Markov processes W_1^N, \dots, W_m^N , where each W_i^N converges to a PDMP W_i . We define $\tau_0^N = 0$ and $\tau_i^N = \tau_{i-1}^N + \mathcal{G}_A(W_i^N)$, and similarly $\tau_0 = 0$ and $\tau_i = \tau_{i-1} + \mathcal{G}_A(W_i)$ for $i = 1, \dots, (m-1)$. The initial states of the processes W_i^N and W_i are given by $W_i^N(0) = f_i(W_{i-1}^N(\tau_{i-1}^N))$ and $W_i(0) = f_i(W_{i-1}(\tau_{i-1}))$ respectively, for

$i = 2, \dots, m$, where $f_i : \mathbb{R}^n \rightarrow \mathbb{R}^n$ are measurable functions. Assuming $W_1^N(0) \Rightarrow W_1(0)$, the process W^N given by

$$W^N(t) = \sum_{i=1}^m W_i^N(t - \tau_{i-1}^N) \mathbb{1}_{\{\tau_{i-1}^N \leq t < \tau_i^N\}} \quad (\text{B5})$$

converges in distribution to the process W defined by

$$W(t) = \sum_{i=1}^m W_i(t - \tau_{i-1}) \mathbb{1}_{\{\tau_{i-1} \leq t < \tau_i\}}. \quad (\text{B6})$$

In our adaptive scheme the times of adaptation are defined analogously to (B1) where the set A is given by (12). We denote the times of adaptation by τ_i^N for $i = 1, \dots, (m-1)$, where $(m-1)$ is the number of adaptations. We define $\tau_0^N = 0$ and $\tau_m^N = \infty$ for convenience. The computation of the scaling parameters $\alpha = (\alpha_1, \dots, \alpha_n)$ and $\beta = (\beta_1, \dots, \beta_{n_R})$ only depends on the state of the simulated process at the time of adaptation. Suppose that this computation is given by functions $L_\alpha, L_\beta : \mathbb{R}^n \rightarrow \mathbb{R}^n$, such that for $\tau_i^N \leq t < \tau_{i+1}$ we have $\alpha(t) = L_\alpha(W^N(\tau_i^N))$ and $\beta(t) = L_\beta(W_i^N(\tau_i^N))$. Here W^N denotes the adaptive version of the scaled family of processes in (3). We can write W^N in the form (B5) with

$$W_i^N(t) = W_i^N(0) + N^{-\alpha(\tau_{i-1}^N)} \sum_{k=1}^{n_R} Y_k \left(\int_0^t N^{\beta(\tau_{i-1}^N) + \alpha(\tau_{i-1}^N) \cdot \nu_k} \lambda_k^N(W_i^N(s)) ds \right) \xi_k$$

where $N^x = \text{diag}(N^{x_1}, \dots, N^{x_k})$ for a k -dimensional vector $x = (x_1, \dots, x_k)$. The initial conditions are given by $W_1^N(0) = N_0^{-\alpha(0)} X(0)$ and $W_i^N(0) = N_0^{\alpha(\tau_{i-1}^N) - \alpha(\tau_i^N)} W_i^N(\tau_i^N)$ for $i = 2, \dots, m$. From Kang et al.¹⁹ we know that each W_i^N converges in distribution to a PDMP W_i , as described in Section II. From the previous lemma it follows that $W^N(t) \Rightarrow W(t)$ for any $t \geq 0$, where W has the form (B6) with the W_i -s given by the convergence result of Kang et al.¹⁹, given in II.

Now we return to our original process X from Section II and recall that for large N_0 we have $X(t) = N_0^\alpha Z^{N_0}(t) \approx_d N_0^\alpha Z(t)$. With adaptation we also have $X(t) = N_0^\alpha W^{N_0}(t)$ and as $W^N(t) \Rightarrow W(t)$ for any $t \geq 0$, it follows that $X(t) = N_0^\alpha W^{N_0}(t) \approx_d N_0^\alpha W(t)$ and this justifies the use of our adaptive scheme to approximate the distribution of the original process X at any fixed time $t \geq 0$.

REFERENCES

- ¹J. Goutsias, "Classical versus stochastic kinetics modeling of biochemical reaction systems," *Biophysical journal*, vol. 92, no. 7, pp. 2350–2365, 2007.

- ²H. H. McAdams and A. Arkin, “Stochastic mechanisms in gene expression,” *Proceedings of the National Academy of Sciences*, vol. 94, no. 3, pp. 814–819, 1997.
- ³B. Munsky, A. Hernday, D. Low, and M. Khammash, “Stochastic modeling of the pap-pili epigenetic switch,” *Proc. FOSBE*, pp. 145–148, 2005.
- ⁴D. Fange and J. Elf, “Noise-induced Min phenotypes in E. coli,” *PLoS computational biology*, vol. 2, no. 6, pp. –80, 2006.
- ⁵T. S. Gardner, C. R. Cantor, and J. J. Collins, “Construction of a genetic toggle switch in Escherichia coli,” *Nature*, vol. 403, pp. 339–42, Jan. 2000.
- ⁶M. B. Elowitz and S. Leibler, “A synthetic oscillatory network of transcriptional regulators,” *Nature*, vol. 403, no. 6767, pp. 335–338, 2000.
- ⁷J. Goutsias and G. Jenkinson, “Markovian dynamics on complex reaction networks,” *Physics Reports*, 2013.
- ⁸B. Munsky and M. Khammash, “The finite state projection algorithm for the solution of the chemical master equation,” *The Journal of chemical physics*, vol. 124, p. 044104, 2006.
- ⁹V. Kazeev, M. Khammash, M. Nip, and C. Schwab, “Direct solution of the Chemical Master Equation using quantized tensor trains,” *PLoS computational biology*, vol. 10, p. e1003359, Mar. 2014.
- ¹⁰D. T. Gillespie, “Stochastic simulation of chemical kinetics,” *Annu. Rev. Phys. Chem.*, vol. 58, pp. 35–55, 2007.
- ¹¹W. E, D. Liu, and E. Vanden-Eijnden, “Nested stochastic simulation algorithm for chemical kinetic systems with disparate rates,” *The Journal of chemical physics*, vol. 123, p. 194107, Nov. 2005.
- ¹²Y. Cao, D. T. Gillespie, and L. R. Petzold, “The slow-scale stochastic simulation algorithm,” *The Journal of chemical physics*, vol. 122, p. 14116, Jan. 2005.
- ¹³D. T. Gillespie, “Approximate accelerated stochastic simulation of chemically reacting systems,” *The Journal of Chemical Physics*, vol. 115, p. 1716, 2001.
- ¹⁴Y. Cao, D. T. Gillespie, and L. R. Petzold, “Efficient step size selection for the tau-leaping simulation method,” *The Journal of chemical physics*, vol. 124, p. 044109, 2006.
- ¹⁵Y. Cao, D. T. Gillespie, and L. R. Petzold, “Adaptive explicit-implicit tau-leaping method with automatic tau selection,” *The Journal of chemical physics*, vol. 126, no. 22, p. 224101, 2007.
- ¹⁶M. Rathinam, L. R. Petzold, Y. Cao, and D. T. Gillespie, “Stiffness in stochastic chemically

- reacting systems: The implicit tau-leaping method,” *The Journal of Chemical Physics*, vol. 119, p. 12784, 2003.
- ¹⁷T. G. Kurtz, “Strong approximation theorems for density dependent Markov chains,” *Stochastic Processes and Their Applications*, vol. 6, no. 3, pp. 223–240, 1978.
- ¹⁸The concentration is the copy-number divided by the volume of the system.
- ¹⁹H.-W. Kang and T. G. Kurtz, “Separation of time-scales and model reduction for stochastic reaction networks,” *The Annals of Applied Probability*, vol. 23, no. 2, pp. 529–583, 2013.
- ²⁰A. Crudu, A. Debussche, and O. Radulescu, “Hybrid stochastic simplifications for multi-scale gene networks,” *BMC systems biology*, vol. 3, no. 1, p. 89, 2009.
- ²¹M. H. Davis, “Piecewise-deterministic Markov processes: A general class of non-diffusion stochastic models,” *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 353–388, 1984.
- ²²E. L. Haseltine and J. B. Rawlings, “Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics,” *The Journal of chemical physics*, vol. 117, no. 15, pp. 6959–6969, 2002.
- ²³J. Pahle, *Eine Hybridmethode zur Simulation biochemischer Prozesse*. PhD thesis, Diploma thesis, ILKD, Universität Karlsruhe (TH) and EML, Heidelberg, 2002.
- ²⁴N. A. Neogi, “Dynamic partitioning of large discrete event biological systems for hybrid simulation and analysis,” in *Hybrid Systems: Computation and Control*, pp. 463–476, Springer, 2004.
- ²⁵M. Bentele and R. Eils, “General stochastic hybrid method for the simulation of chemical reaction processes in cells,” in *Computational Methods in Systems Biology*, pp. 248–251, Springer, 2005.
- ²⁶H. Salis and Y. Kaznessis, “Accurate hybrid stochastic simulation of a system of coupled chemical or biochemical reactions,” *The Journal of chemical physics*, vol. 122, p. 054103, 2005.
- ²⁷A. Crudu, A. Debussche, A. Muller, and O. Radulescu, “Convergence of stochastic gene networks to hybrid piecewise deterministic processes,” *The Annals of Applied Probability*, vol. 22, no. 5, pp. 1822–1859, 2012.
- ²⁸A. Alfonsi, E. Cancès, G. Turinici, B. Di Ventura, and W. Huisinga, “Adaptive simulation of hybrid stochastic and deterministic models for biochemical systems,” in *ESAIM: Proceedings*, vol. 14, pp. 1–13, EDP Sciences, 2005.

- ²⁹S. Hoops, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer, "COPASI - A COmplex PAthway SIMulator," *Bioinformatics*, vol. 22, pp. 3067–3074, 2006.
- ³⁰M. Griffith, T. Courtney, J. Peccoud, and W. H. Sanders, "Dynamic partitioning for hybrid simulation of the bistable HIV-1 transactivation network.," *Bioinformatics (Oxford, England)*, vol. 22, pp. 2782–9, Nov. 2006.
- ³¹J. Puchaka and A. M. Kierzek, "Bridging the gap between stochastic and deterministic regimes in the kinetic simulations of the biochemical reaction networks.," *Biophysical journal*, vol. 86, pp. 1357–72, Mar. 2004.
- ³²K. Burrage, T. Tian, and P. Burrage, "A multi-scaled approach for simulating chemical reaction systems.," *Progress in biophysics and molecular biology*, vol. 85, no. 2-3, pp. 217–34, 2004.
- ³³L. a. Harris and P. Clancy, "A "partitioned leaping" approach for multiscale modeling of chemical reaction dynamics.," *The Journal of chemical physics*, vol. 125, p. 144107, Oct. 2006.
- ³⁴J. R. Chubb, T. Trcek, S. M. Shenoy, and R. H. Singer, "Transcriptional pulsing of a developmental gene," *Current biology*, vol. 16, no. 10, pp. 1018–1025, 2006.
- ³⁵J. Pahle, "Biochemical simulations: stochastic, approximate stochastic and hybrid approaches.," *Briefings in bioinformatics*, vol. 10, pp. 53–64, Jan. 2009.
- ³⁶C. V. Rao and A. P. Arkin, "Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm," *The Journal of Chemical Physics*, vol. 118, no. 11, p. 4999, 2003.
- ³⁷IUPAC. Compendium of Chemical Terminology, 2nd ed. (the *Gold Book*). Compiled by A. D. McNaught and A. Wilkinson. Blackwell Scientific Publications, Oxford (1997). XML on-line corrected version: <http://goldbook.iupac.org> (2006-) created by M. Nic, J. Jirat, B. Kosata; updates compiled by A. Jenkins. ISBN 0-9678550-9-8. doi:10.1351/goldbook.
- ³⁸D. T. Gillespie, "A diffusional bimolecular propensity function," *The Journal of chemical physics*, vol. 131, no. 16, pp. 164109–164109, 2009.
- ³⁹S. N. Ethier and T. G. Kurtz, *Markov processes: characterization and convergence*, vol. 282. Wiley. com, 2009.
- ⁴⁰S. Peleš, B. Munsky, and M. Khammash, "Reduction and solution of the chemical master equation using time scale separation and finite state projection," *The Journal of chemical*

- physics*, vol. 125, no. 20, p. 204104, 2006.
- ⁴¹O. Radulescu, A. N. Gorban, A. Zinovyev, and A. Lilienbaum, “Robust simplifications of multiscale biochemical networks,” *BMC systems biology*, vol. 2, no. 1, p. 86, 2008.
- ⁴²The implementation allows different ODE solvers to be used. However, for the examples we used the Dormand-Prince 5 stepper from the boost odeint library (<http://www.boost.org/>). The boost odeint library is coupled to our implementation via a Java Native Interface wrapper.
- ⁴³For subnetworks that are weakly reversible and have deficiency zero, we compute the moments of the product-form stationary distribution, where all the species satisfy a conservation relation, by approximating it with a multinomial distribution with appropriately defined virtual species. The error incurred by this approximation is generally small enough to be neglected. If no species satisfy a conservation relation, then the stationary distribution is just the product of independent Poisson distributions.
- ⁴⁴J. Elf and M. Ehrenberg, “Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases,” *Systems biology*, vol. 1, no. 2, pp. 230–236, 2004.
- ⁴⁵E. Seneta, *Non-negative Matrices and Markov Chains (Springer Series in Statistics)*. Springer, 2006.
- ⁴⁶D. F. Anderson, G. Craciun, and T. G. Kurtz, “Product-form stationary distributions for deficiency zero chemical reaction networks,” *Bulletin of mathematical biology*, vol. 72, no. 8, pp. 1947–1970, 2010.
- ⁴⁷A SRN is called *weakly reversible* if for every reaction k there is a sequence of reactions $\nu_{k_1} \rightarrow \nu'_{k_1} \rightarrow \dots \rightarrow \nu_{k_m} \rightarrow \nu'_{k_m}$ such that $\nu'_k = \nu_{k_1}$ and $\nu_k = \nu'_{k_m}$.
- ⁴⁸The *deficiency* of a SRN is defined as $\delta = |C| - l - s$ where $|C|$ is the number of reaction complexes ($C = \{\nu_k\} \cap \{\nu'_k\}$), l is the number of linkage classes (a linkage class is a connected component of the reaction complex graph corresponding to the SRN) and s is the dimension of the stoichiometric subspace $S = \text{span}_{k \in \{1, \dots, n_R\}} \{\xi_k\}$. We refer the readers to⁴⁶ for more details.
- ⁴⁹A. Gupta and M. Khammash, “Determining the long-term behavior of cell populations: A new procedure for detecting ergodicity in large stochastic reaction networks,” *arXiv preprint arXiv:1312.2879*, 2013.
- ⁵⁰J. R. Baxter and R. V. Chacon, “Compactness of stopping times,” *Probability Theory and Related Fields*, vol. 40, no. 3, pp. 169–181, 1977.